Università degli studi di Roma
Tor Vergata

Tesi di Dottorato di Ricerca in
Informatica e Ingegneria dell'Automazione

Complex Networks: Analysis and Control

Claudio Argento

Roma, Giugno 2007

Complex Networks: Analysis and Control

Complex Networks: Analysis and Control

Claudio Argento

Advisor
Prof. Antonio Tornambè

Co-advisor
Prof. Chaouki T. Abdallah

Coordinator
Prof. Daniel P. Bovet

Acknowledgments

I wish to thank all the people whose behavior had some positive impact on the present work. It is probably impossible to mention all of them, because I would inevitably forget some name and also because it would require one or more chapters, not just a simple section. For this reason, I apologize to all the people who are not mentioned in this section, though they should have been.

The first thank is for my family, who supported me during the years of PhD studies in any possible way. It is pointless to say that their contribution has been determinant.

I thank my advisor Prof. Antonio Tornambè, who helped me to deepen my knowledge on Control Theory and Robotics and guided me during the preparation of this work. I also thank Prof. Laura Menini, who helped me to deepen my knowledge on System Theory. I thank my co-advisor Prof. Chaouki T. Abdallah for introducing me to the study of complex networks and also the Department of Electrical and Computer Engineering at the University of New Mexico for the period I spent there as research scholar. In particular, I wish to thank Prof. Peter Dorato and Ms. Mimi Stephens: the pleasant Italian conversations I had with them made me feel "closer" to home. I also wish to thank all the people I knew in New Mexico during my stay for their help and kindness. I thank Sima Azar, Joud Khoury and Nicolas Nehme Antoun, with who I shared the office at UNM. A special thank is for Nathan Menhorn, for his help and for the fun we had visiting New Mexico.

I thank all the professors and researchers at DISP at Tor Vergata, who in-

fluenced in some way this period of my life: Prof. Alessandro Astolfi, Dr. Sergio Galeani, Prof. Osvaldo Maria Grasselli, Dr. Francesco Martinelli, Prof. Salvatore Nicosia and Prof. Luca Zaccarian. A special thank is for Dr. Sergio Galeani, with who I had the pleasure to share the office during the first year of my PhD; he, like Prof. Laura Menini, helped me in deepening my knowledge on System Theory and was always kind and helpful with me.

I wish to thank all my PhD colleagues: Daniele Carnevale, Fulvio Forni, Simona Onori, Luigi Pangione, Serena Pani, Fabio Piedimonte, Alessandro Potini and Giuseppe Viola. I also thank Ciro Collaro and Claudio Cosentino, who shared with me and Simona the nice experience at the University of New Mexico.

Finally, I thank all the people that during these years have supported me, even telling me just a nice word.

Contents

Chapter 1

Introduction to complex networks

This chapter aims to provide an overview on complex networks, trying to investigate what apparently different kinds of networks have in common. Systems belonging to different branches of science, such as metabolic pathways and ecosystems, the Internet and propagation of HIV infection share similar network architectures. Many systems in the real-world can be modeled as networks. For example, the patterns of connections between economic agents form a social network, since often partners are chosen not on economic grounds but for social reasons. It is clear that the structure of such networks affects the pattern of economic transactions. In the last years, researchers have conducted extensive investigations of networks in economics, mathematics, sociology and several other fields, trying to understand and explain network effects [93]. Inspired by empirical studies of networked control systems such as the Internet, social networks and biological networks, researchers have developed a variety of techniques and models to help us understand or predict the behavior of these systems [92]. The content of this chapter is mainly based on [12, 20, 74, 91, 92, 95, 126]. My research results are presented in Sec. 1.6, Chap. 2 and Chap. 3.

1.1 Introduction

A network is a set of items, called vertices or sometimes nodes, with connections between them, called edges (Fig. 1.1). Examples of systems taking the

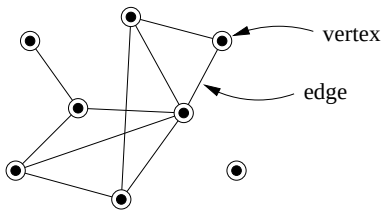

Figure 1.1: A small example network with eight vertices and ten edges [92].

form of networks (also called graphs in much of the mathematical literature) are ubiquitous in the real-world: the Internet, the World Wide Web, social networks of acquaintance or other connections between individuals, neural networks, metabolic networks, food webs, networks of citations between papers, etc (Fig. 1.2).

The first proof in theory of networks is probably the Euler's solution of the Konïgsberg bridge problem. Networks have been studied extensively in the social sciences: in particular, typical social network studies address issues of centrality (which individuals are best connected to others or have most influence) and connectivity (wether and how individuals are connected to one another through the network).

Recently, thanks to the availability of computers and communication networks, the focus of network research is shifting away from the analysis of single small graphs and the properties of individual vertices or edges within such graphs to consideration of large-scale statistical properties of graphs. This change of scale implies a corresponding change in our analytic approach. For example, with reference to a network containing millions of vertices, it has little meaning asking which vertex would prove most crucial to the network connectivity if it were removed. On the other hand, it is meaningful asking what percentage of vertices need to be removed to substantially affect

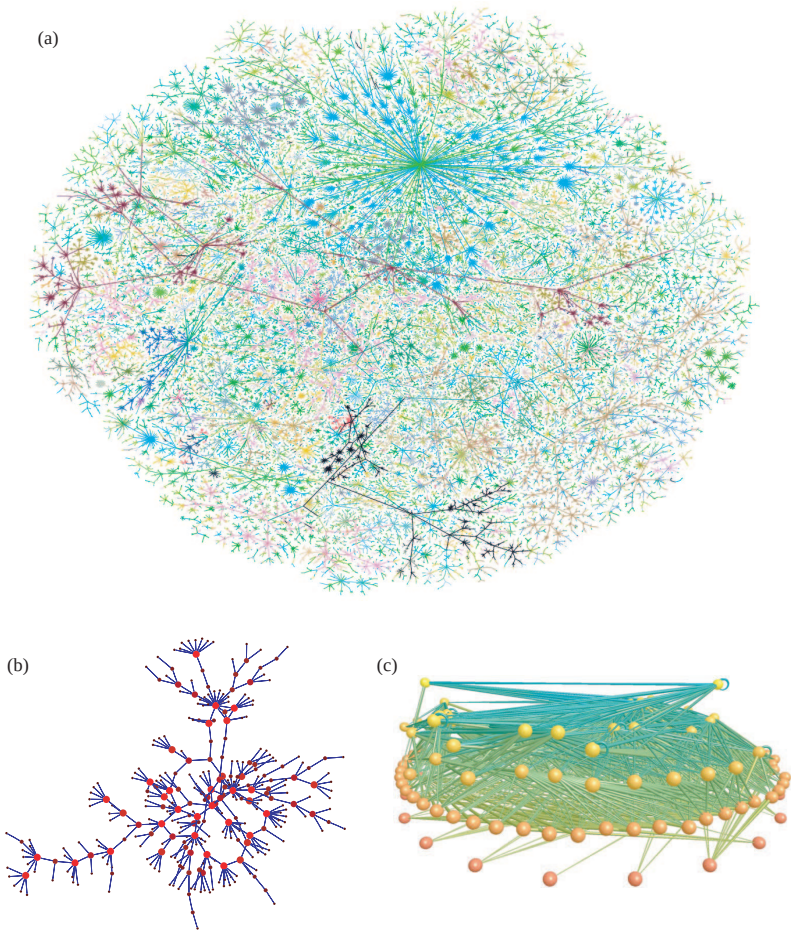

Figure 1.2: Three examples of real-world networks. (a) A visualization of the network structure of the Internet at the level of "autonomous systems" – local groups of computers each representing hundreds or thousands of machines. Picture by Hal Burch and Bill Cheswick (b) A social network, in this case of sexual contacts, redrawn from the HIV data of Potterat et al. [101]. (c) A food web of predator-prey interactions between species in a freshwater lake [81].

network connectivity. Moreover, the analysis of large networks must necessarily abstract from their graphical representation: the role played in the past by the human eye is now played by the statistical methods, which permit to investigate the structure and properties of very large networks. In fact, for networks of tens or hundreds of vertices, it is a relatively straightforward matter to draw a picture of the network with actual points and lines, and to answer specific questions about network structure by examining this picture. This has been one of the primary methods of network analysts since the field began (Fig. 1.3). The human eye is an analytic tool of remarkable power, and

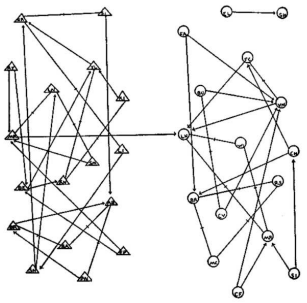

Figure 1.3: An early hand-drawn social network from 1934 representing friendships between school children. After Moreno [85].

eyeballing pictures of networks is an excellent way to gain an understanding of their structure. With a network of a million or a billion vertices, however, this approach is useless. (Fig. 1.2a shows an example of a network that lies at the upper limit of what can usefully be drawn on a piece of paper or computer screen.)

Researchers have got important results on the characterization and modeling of network structure, while a lot of work still remains to do for understanding the effects of structure on system behavior.

A network can be mathematically represented by a graph [116] in which vertices represent systems and edges represent the relationships between them. There can be more than one type of vertex or edge and vertices or edges may

have a variety of properties associated with them. Graphs can be weighted or unweighted, directed or undirected; directed graphs (or digraphs, for short) can be cyclic or acyclic, etc (Fig. 1.4). Graphs containing hyperedges (edges

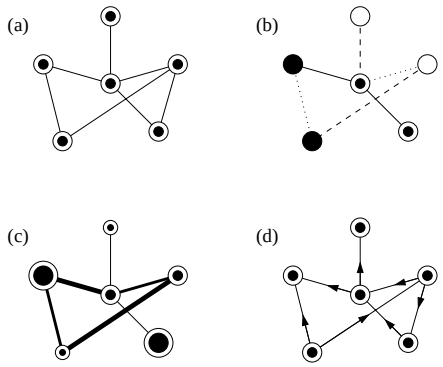

Figure 1.4: Examples of various types of networks [92]: (a) an undirected network with only a single type of vertex and a single type of edge; (b) a network with a number of discrete vertex and edge types; (c) a network with varying vertex and edge weights; (d) a directed network in which each edge has a direction.

joining more than two vertices together) are called hypergraphs. Graphs that contain vertices of two distinct types, with edges running only between unlike types, are called bipartite graphs (Fig. 1.5). An example of bipartite graph are the affiliation (or collaboration) networks, in which people are joined together by common membership of groups, the two types of vertices representing the people and the groups. An example of such networks is the network of collaborations of movie actors: in this case, the two types of vertices are movies and actors, and the network can be represented as a graph with edges running between each movie and the actors that appear in it. Researchers have also considered the projection of this graph onto the unipartite space of actors only, also called a one-mode network [123] (Fig. 1.5). In such a projection two actors are considered connected if they have appeared in a movie together. The construction of the one-mode network however involves discarding some

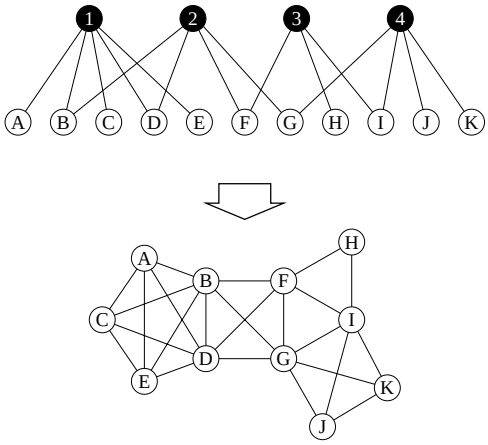

Figure 1.5: A schematic representation (top) of a bipartite graph, such as the graph of movies and the actors who have appeared in them. In this small graph there are four movies, labeled 1 to 4, and eleven actors, labeled A to K, with edges joining each movie to the actors in its cast. The lower part of the picture shows the one-mode projection of the graph for the eleven actors [95].

of the information contained in the original bipartite network, and for this reason it is more desirable to model collaboration networks using the full bipartite structure. Moreover, graphs may evolve over time, with vertices or edges appearing or disappearing, or values defined on those vertices and edges changing.

The jargon of the study of networks is unfortunately confused by differing usages among investigators from different fields. To avoid (or at least reduce) confusion, a short glossary of terms as they are used in this thesis is provided:

- Vertex (pl. vertices): The fundamental unit of a network, also called a site (physics), a node (computer science), or an actor (sociology).

- Edge: The line connecting two vertices. Also called a bond (physics), a link (computer science), or a tie (sociology).

- Directed/undirected: An edge is directed if it runs in only one direction

(such as a one-way road between two points), and undirected if it runs in both directions. Directed edges, which are sometimes called arcs, can be thought of as sporting arrows indicating their orientation. A graph is directed if all of its edges are directed. An undirected graph can be represented by a directed one having two edges between each pair of connected vertices, one in each direction.

- Degree: The number of edges connected to a vertex. Note that the degree is not necessarily equal to the number of vertices adjacent to a vertex, since there may be more than one edge between any two vertices. In a few recent articles, the degree is referred to as the "connectivity" of a vertex, but here this usage is avoided because the word connectivity already has another meaning in graph theory. A directed graph has both an in-degree and an out-degree for each vertex, which are the numbers of incoming and outgoing edges respectively.

- Component: The component to which a vertex belongs is that set of vertices that can be reached from it by paths running along edges of the graph. In a directed graph a vertex has both an in-component and an out-component, which are the sets of vertices from which the vertex can be reached and which can be reached from it.

- Geodesic path: A geodesic path is the shortest path through the network from one vertex to another. Note that there may be and often is more than one geodesic path between two vertices.

- Diameter: The diameter of a network is the length (in number of edges) of the longest geodesic path between any two vertices. A few authors have also used this term to mean the average geodesic distance in a graph, although strictly the two quantities are quite distinct.

1.2 Real-world networks

A social network is a set of people or groups of people with some pattern of contacts or interactions between them [107, 123] (Fig. 1.2b). The first theorist to stress the importance of the degree distribution (i.e., the probability with which a vertex has a certain number of edges connected to it) in networks of all kinds was probably Anatol Rapoport [102], whose mathematical models are of particular note. Another important set of experiments are the famous "small-world" experiments of Milgram [82, 119], which probed the distribution of path lengths in an acquaintance network by asking participants to pass a letter to one of their first-name acquaintances in an attempt to get it to an assigned target individual. Most of the letters in the experiment were lost, but about a quarter reached the target and passed on average through the hands of only about six people in doing so. This experiment was the origin of the popular concept of the "six degrees of separation", although that phrase did not appear in Milgram's writing, being coined some decades later by Guare [54].

Traditional social network studies often suffer from problems of inaccuracy, subjectivity and small sample size. Because of these problems, many researchers have turned to other methods for probing social networks. One source of copious and relatively reliable data is collaboration networks. These are typically affiliation networks in which participants collaborate in groups of one kind or another and links between pairs of individuals are established by common group membership. Examples of such networks are the collaboration network of film actors, networks of company directors, networks of coauthorship among academics, etc.

The classic example of an information network is the network of citations between academic papers [44]. These citations form a network in which the vertices are articles and a directed edge from article A to article B indicates that A cites B (Fig. 1.6). Citation networks are acyclic because papers can only cite other papers that have already been written, not those that have yet

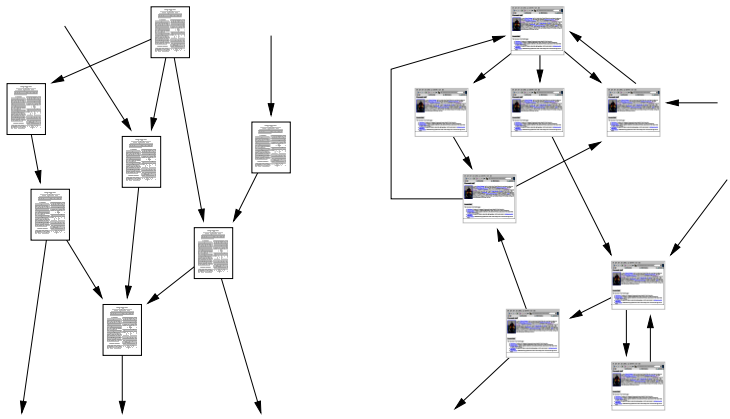

Figure 1.6: The two best studied information networks [92]. Left: the citation network of academic papers in which the vertices are papers and the directed edges are citations of one paper by another. Since papers can only cite those that came before them (lower down in the figure) the graph is acyclic – it has no closed loops. Right: the World Wide Web, a network of text pages accessible over the Internet, in which the vertices are pages and the directed edges are hyperlinks. There are no constraints on the Web that forbid cycles and hence it is in general cyclic.

to be written. In 1926 Alfred Lotka discovered the so-called Law of Scientific Productivity, which states that the distribution of the numbers of papers written by individual scientists follows a power law: the number of scientists who have written k papers falls off as $k^{-\alpha}$ for some constant α. An interesting development in the study of citation patterns has been the arrival of automatic citation "crawlers" that construct citation networks from online papers.

Another important example of an information network is the World Wide Web, which is a network of Web pages containing information, linked together by hyperlinks from one page to another [56] (Fig. 1.6). The Web should not be confused with the Internet, which is a physical network of computers linked together by optical fibre and other data connections. Unlike a citation network, the World Wide Web is cyclic: there is no natural ordering of sites and no constraints that prevent the appearance of closed loops. The Web is a directed network and appears to have power-law distributions for both in- and out-degree. Data about the Web come from "crawls" of the network, in which Web pages are found by following hyperlinks from other pages [24]. The obtained picture of the network structure of the World Wide Web is therefore necessarily biased: crawls usually cover only a part of the Web, thus pages are more likely to be found the more other pages point to them [76].

Preference networks provide an example of a bipartite information network. A preference network is a network with two kinds of vertices representing individuals and the objects of their preference, such as books or films, with an edge connecting each individual to the books or films they like. Preference networks can be weighted to indicate strength of likes or dislikes and can also be thought of as social networks, linking not only people to objects, but also people to other people with similar preferences. Networks of this kind form the basis for collaborative filtering algorithms and recommender systems, which are techniques for predicting new likes or dislikes based on comparison of individuals' preferences with those of others. Collaborative filtering has found considerable commercial success for product recommendation and targeted

advertising particularly with online retailers.

Technological networks are man-made networks designed typically for distribution of some commodity or resource, such as electricity or information. Examples of distribution networks are the electric power grid [14, 124, 126], the network of airline routes [14], and networks of roads [64], railways [73, 109] and pedestrian traffic [28].

Another widely studied technological network is the Internet, i.e., the network of physical connections between computers. Since there is a large and ever-changing number of computers on the Internet, the structure of the network is usually examined at a coarse-grained level, either the level of routers, special-purpose computers on the network that control the movement of data, or autonomous systems, which are groups of computers within which networking is handled locally, but between which data flows over the public Internet (Fig. 1.2a). Traceroute programs can report the sequence of network nodes that a data packet passes through when traveling between two points and are therefore used to reconstruct the physical structure of the network.

The classic example of a biological network is probably the network of metabolic pathways, which is a representation of metabolic substrates and products with directed edges joining them if a known metabolic reaction exists that acts on a given substrate and produces a given product. A separate network is the network of mechanistic physical interactions between proteins (as opposed to chemical reactions among metabolites), which is usually referred to as a protein interaction network.

Another example of a biological network is the food web (Fig. 1.2c), in which the vertices represent species in an ecosystem and a directed edge from species A to species B indicates that A preys on B. Sometimes the relationship is drawn the other way around, because ecologists tend to think in terms of energy or carbon flows through food webs; a predator-prey interaction is thus drawn as an arrow pointing from prey to predator, indicating energy flow from prey to predator when the prey is eaten. Neural networks, blood vessels and

the equivalent vascular networks in plants are other examples of biological networks.

1.3 Properties of networks

The famous experiments carried out by Stanley Milgram in the 1960s, described in Sec. 1.2, are one of the first direct demonstrations of the small-world effect, the fact that most pairs of vertices in most networks seem to be connected by a short path through the network. The existence of the small-world effect had been speculated upon before Milgram's work, notably in a remarkable 1929 short story by the Hungarian writer Frigyes Karinthy [65] and more rigorously in the mathematical work of Pool and Kochen [36] which, although published after Milgram's studies, was in circulation in preprint form for a decade before Milgram took up the problem.

1.3.1 Characteristic path length and clustering coefficient

For an undirected and unweighted (topological) network, the average distance between vertex pairs (characteristic path length) is defined as [72, 74]

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

where N is the number of nodes and d_{ij} is the geodesic (shortest) distance from vertex i to vertex j. In other words, L is the number of edges in the shortest path between two vertices, averaged over all pairs of vertices. It has been shown that many real-world networks exhibit small values of L, much smaller than the number N of vertices.

If the graph is not connected, there exist vertex pairs that have no connecting path or, equivalently, whose geodesic distance is infinite and, accordingly, L becomes infinite. To avoid this problem, L is usually defined on such networks to be the mean geodesic distance between all pairs that have a connecting

path. Pairs that fall in two different components (maximal subsets of vertices that are connected by paths through the network) are excluded from the average.

The small-world effect has obvious implications for the dynamics of processes taking place on networks, as for example the spread of information across a network, and also underlies some well-known parlor games, particularly the calculation of Erdős numbers [35] and Bacon numbers.

On the other hand, the small-world effect is also mathematically obvious: in fact, if the number n of vertices within a distance r of a typical central vertex grows exponentially with r – and this is true of many networks, including the random graph – then the value of L will increase as $\ln N$:

$$n \propto e^{\alpha r} \Rightarrow N \propto e^{\alpha \frac{d}{2}} \Rightarrow d \propto \ln N \Rightarrow L \propto \ln N \,,$$

where α is a positive real constant and d is the diameter of the network, that is, the length (in number of edges) of the longest geodesic path between any two vertices. Networks are said to show the small-world effect if the value of L scales logarithmically or slower with network size for fixed mean degree [92]. Logarithmic scaling has been observed in various real-world networks [9, 88, 89].

In many networks it is found that if vertex A is connected to vertex B and vertex B to vertex C, then there is a heightened probability that vertex A will also be connected to vertex C. In the language of social networks, the friend of your friend is likely also to be your friend. This property is called network transitivity. In terms of network topology, transitivity means the presence of a heightened number of triangles in the network – sets of three vertices each of which is connected to each of the others. It can be quantified by defining a clustering coefficient C thus:

$$C = \frac{3 \times \text{number of triangles in the network}}{\text{number of connected triples of vertices}} \,, \tag{1.1}$$

where a "connected triple" means a single vertex with edges running to an unordered pair of others (Fig. 1.7). In effect, C measures the fraction of

triples that have their third edge filled in to complete the triangle. The factor of three in the numerator accounts for the fact that each triangle contributes to three triples and ensures that C lies in the range $0 \leq C \leq 1$. In simple terms, C is the mean probability that two vertices that are network neighbors of the same other vertex will themselves be neighbors.

An alternative definition of the clustering coefficient has been given by Watts and Strogatz [126], who proposed defining a local value

$$C_i = \frac{\text{number of triangles connected to vertex } i}{\text{number of triples centered on vertex } i}\,. \tag{1.2}$$

For vertices with degree 0 or 1, for which both numerator and denominator are zero, $C_i := 0$. The clustering coefficient for the whole network is the average

$$C = \frac{1}{N}\sum_i C_i\,. \tag{1.3}$$

The quantities in (1.2) and (1.3) can also be computed in the following way. If the generic vertex i has k_i neighbors, then at most $k_i(k_i-1)/2$ edges can exist between them (this occurs when every neighbor of i is connected to every other neighbor of i); C_i is the fraction of these allowable edges that actually exist and C is the average of C_i over all i. Fig. 1.7 illustrates the difference

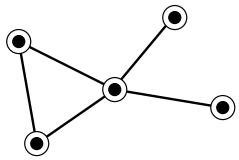

Figure 1.7: Illustration of the definition of the clustering coefficient C (1.1) [92]. This network has one triangle and eight connected triples, and therefore has a clustering coefficient of $3 \times 1/8 = \frac{3}{8}$. The individual vertices have local clustering coefficients (1.2) of 1, 1, $\frac{1}{6}$, 0 and 0, for a mean value (1.3) of $C = \frac{13}{30}$.

between the two definitions of the clustering coefficient.

The clustering coefficient measures the density of triangles in a network. An obvious generalization is to ask about the density of longer loops also: loops of length four and above. If more than one edge is permitted between a pair of vertices, then there is also a lower order clustering coefficient that describes the density of loops of length two. This coefficient is particularly important in directed graphs where the two edges in question can point in opposite directions. The probability that two vertices in a directed network point to each other is called the reciprocity and is often measured in directed social networks [107, 123].

1.3.2 Networks between regularity and randomness

Ordinarily, the connection topology of several network models is assumed to be either completely regular or completely random. But many biological, technological and social networks lie somewhere between these two extremes. In [126], the authors explore simple models of networks that can be tuned through this middle ground: regular networks 'rewired' to introduce increasing amounts of disorder. They find that these systems can be highly clustered, like regular lattices, yet have small characteristic path lengths, like random graphs. They call them 'small-world' networks, by analogy with the small-world phenomenon [54, 70, 82]. The neural network of the worm Caenorhabditis elegans, the power grid of the western United States and the collaboration graph of film actors are examples of small-world networks.

To interpolate between regular and random networks, the following random rewiring procedure is considered (Fig. 1.8). Starting from a ring lattice with N vertices and k edges per vertex, each edge is randomly rewired with probability p. This construction allow to 'tune' the graph between regularity ($p = 0$) and disorder ($p = 1$), and thereby to probe the intermediate region $0 < p < 1$, about which little is known. A vertex and the edge that connects it to its nearest neighbor are chosen in a clockwise sense. The edge is reconnected with probability p to a vertex chosen uniformly at random over the entire

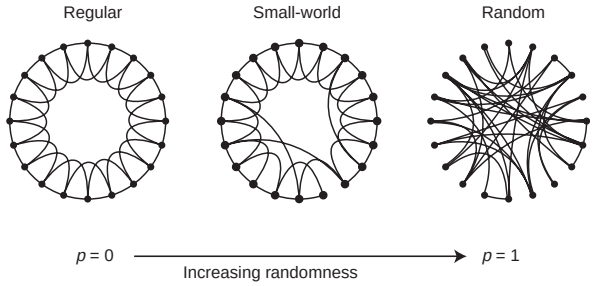

Figure 1.8: Random rewiring procedure for interpolating between a regular ring lattice and a random network, without altering the number of vertices or edges in the graph ($N = 20$, $k = 4$) [126].

ring, with duplicate edges forbidden; otherwise the edge is left in place. This process is repeated by moving clockwise around the ring, considering each vertex in turn until one lap is completed. Next, the edges that connect vertices to their second-nearest neighbors are considered clockwise. As before, each of these edges is randomly rewired with probability p, and this process continues, circulating around the ring and proceeding outwards to more distant neighbors after each lap, until each edge in the original lattice has been considered once. (As there are $Nk/2$ edges in the entire graph, the rewiring process stops after $k/2$ laps.)

Three realizations of this process are shown in Fig. 1.8, for different values of p. For $p = 0$, the original ring is unchanged; as p increases, the graph becomes increasingly disordered until for $p = 1$ all edges are rewired randomly. For intermediate values of p, the graph is a small-world network: highly clustered like a regular graph, yet with small characteristic path length, like a random graph (Fig. 1.9). The structural properties of these graphs are quantified by their characteristic path length $L(p)$ and clustering coefficient $C(p)$. In particular, $L(p)$ measures the typical separation between two vertices in the graph (a global property), whereas $C(p)$ measures the cliquishness of a typical neighborhood (a local property). The considered networks have many vertices with sparse connections, but not so sparse that the graph is in danger of be-

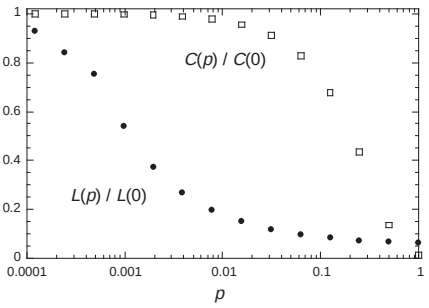

Figure 1.9: Characteristic path length $L(p)$ and clustering coefficient $C(p)$ for the family of randomly rewired graphs described in Fig. 1.8 [126].

coming disconnected. Specifically, the authors require $N \gg k \gg \ln N \gg 1$, where $k \gg \ln N$ guarantees that a random graph will be connected [23]. In this regime, it is found that $L \sim N/2k \gg 1$ and $C \sim 3/4$ as $p \to 0$, while $L \approx L_{random} \sim \ln N / \ln k$ and $C \approx C_{random} \sim k/N \ll 1$ as $p \to 1$. Thus the regular lattice at $p = 0$ is a highly clustered, large world where L grows linearly with N, whereas the random network at $p = 1$ is a poorly clustered, small world where L grows only logarithmically with N. These limiting cases might lead one to suspect that large C is always associated with large L, and small C with small L. On the contrary, Fig. 1.9 reveals that there is a broad interval of p over which $L(p)$ is almost as small as L_{random} yet $C(p) \gg C_{random}$. These small-world networks result from the immediate drop in $L(p)$ caused by the introduction of a few long-range edges. Such 'short cuts' connect vertices that would otherwise be much farther apart than L_{random}. For small p, each short cut has a highly nonlinear effect on L, contracting the distance not just between the pair of vertices that it connects, but between their immediate neighborhoods, neighborhoods of neighborhoods and so on. By contrast, an edge removed from a clustered neighborhood to make a short cut has, at most, a linear effect on C; hence $C(p)$ remains practically unchanged for small p even though $L(p)$ drops rapidly. The important implication here is that at the local level (as reflected by $C(p)$), the transition to a small world is almost

undetectable. To check the robustness of these results, the authors have tested many different types of initial regular graphs, as well as different algorithms for random rewiring, and all give qualitatively similar results. The only requirement is that the rewired edges must typically connect vertices that would otherwise be much farther apart than L_{random}.

The idealized construction above reveals the key role of short cuts. It suggests that the small-world phenomenon might be common in sparse networks with many vertices, as even a tiny fraction of short cuts would suffice. To test this idea, the authors computed L and C for the collaboration graph of actors in feature films, the electrical power grid of the western United States and the neural network of the nematode worm Caenorhabditis elegans [1]. All three graphs are of scientific interest. The graph of film actors is a surrogate for a social network [123], with the advantage of being much more easily specified. It is also akin to the graph of mathematical collaborations centered, traditionally, on P. Erdős. The graph of the power grid is relevant to the efficiency and robustness of power networks [99] and C. elegans is the sole example of a completely mapped neural network.

For friendship networks, the characteristic path length and the clustering coefficient have intuitive meanings: L is the average number of friendships in the shortest chain connecting two people; C_i reflects the extent to which friends of i are also friends of each other; and thus C measures the cliquishness of a typical friendship circle. The data shown in Fig. 1.9 are averages over 20 random realizations of the rewiring process described in Fig. 1.8, and have been normalized by the values $L(0)$, $C(0)$ for a regular lattice. All the graphs have $N = 1000$ vertices and an average degree of $k = 10$ edges per vertex. A logarithmic horizontal scale has been used to resolve the rapid drop in $L(p)$, corresponding to the onset of the small-world phenomenon. During this drop, $C(p)$ remains almost constant at its value for the regular lattice, indicating that the transition to a small world is almost undetectable at the local level.

The graphs considered in Table 1.1 are defined as follows [126]. Two actors

Table 1.1: Empirical examples of small-world networks [126].

	L_{actual}	L_{random}	C_{actual}	C_{random}
Film actors	3.65	2.99	0.79	0.00027
Power grid	18.7	12.4	0.080	0.005
C. elegans	2.65	2.25	0.28	0.05

are joined by an edge if they have acted in a film together. The attention is restricted to the giant connected component [23] of this graph, which includes $\sim 90\%$ of all actors listed in the Internet Movie Database, as of April 1997. For the power grid, vertices represent generators, transformers and substations, and edges represent high voltage transmission lines between them. For C. elegans, an edge joins two neurons if they are connected by either a synapse or a gap junction. All edges are treated as undirected and unweighted, and all vertices as identical, though these are crude approximations. All three networks show the small-world phenomenon: $L \gtrsim L_{random}$ but $C \gg C_{random}$. These examples were not hand-picked; they were chosen because of their inherent interest and because complete wiring diagrams were available. Thus the small-world phenomenon is not merely a curiosity of social networks nor an artefact of an idealized model – it is probably generic for many large, sparse networks found in nature.

In [126] the functional significance of small-world connectivity for dynamical systems is also investigated. The considered test case is a deliberately simplified model for the spread of an infectious disease. The population structure is modeled by the family of graphs described in Fig. 1.8. At time $t = 0$, a single infective individual is introduced into an otherwise healthy population. Infective individuals are removed permanently (by immunity or death) after a period of sickness that lasts one unit of dimensionless time. During this time, each infective individual can infect each of its healthy neighbors with probability r. On subsequent time steps, the disease spreads along the edges of the graph until it either infects the entire population, or it dies out, having

infected some fraction of the population in the process.

Two results emerge. First, the critical infectiousness r_{half}, at which the disease infects half the population, decreases rapidly for small p (Fig. 1.10a). Second, for a disease that is sufficiently infectious to infect the entire popu-

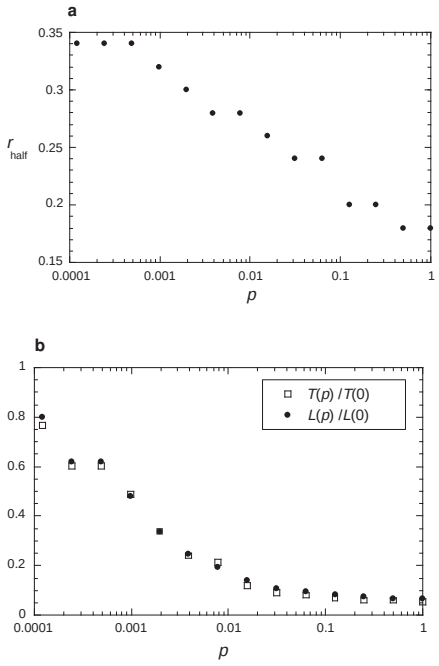

Figure 1.10: The community structure is given by one realization of the family of randomly rewired graphs used in in Fig. 1.8. **a**, Critical infectiousness r_{half}, at which the disease infects half the population, decreases with p. **b**, The time $T(p)$ required for a maximally infectious disease ($r = 1$) to spread throughout the entire population has essentially the same functional form as the characteristic path length $L(p)$. Even if only a few per cent of the edges in the original lattice are randomly rewired, the time to global infection is nearly as short as for a random graph [126].

lation regardless of its structure, the time $T(p)$ required for global infection resembles the $L(p)$ curve (Fig. 1.10b). Thus, infectious diseases are predicted to spread much more easily and quickly in a small world; the alarming and

less obvious point is how few short cuts are needed to make the world small.

1.3.3 Global and local efficiency

The approach of Watts and Strogatz described in Sec. 1.3.2 can be used when the only information retained of a real network is about the existence or the absence of a link, nothing is known about the physical length of the link (or more generically the weight associated with the link, i.e., the graph is unweighted) and multiple edges between the same couple of nodes are not allowed (i.e., the graph is simple). Moreover, the assumption of connectedness is necessary because otherwise the quantity L would diverge.

Of course, a generalization of the approach of Watts and Strogatz to weighted networks would allow a more detailed analysis of real networks and would extend the range of applications. For example, with reference to the same three real networks studied in [126], the analysis of the network of film actors must be restrained to only a part of the system, the giant connected component of the graph, in order to avoid the divergence of L. Moreover, the topological approximation only provides whether actors participated in some movie together or if they did not at all. In reality there are, instead, various degrees of correlation: two actors that have done ten movies together are in a much stricter relation than two actors that have acted together only once. It is possible to better shape this different degree of friendship by using a non-simple graph or by using a weighted network: if two actors have acted together, a weight is associated with their connection by saying that the length of the connection, instead of being always equal to one, is equal to the inverse of the number of movies they did together. In the case of the neural network of the C. elegans, Watts and Strogatz define an edge in the graph when two vertices are connected by either a synapse or a gap junction. This is only a first approximation of the real network. Neurons are different one from the other and some of them are in much stricter relation than others: the number of junctions connecting a couple of neurons can vary a lot, up to

a maximum of 72 in the case of the C. elegans. As in the case of film actors, a weighted network is more suited to describe such a system and can be defined by setting the length of the connection $i - j$ as equal to the minimum between 1 and the inverse of the number of junctions between i and j. The last network studied in [126], the electrical power grid of the western United States, is clearly a network where the geographical distances play a fundamental role. Any of the high voltage transmission lines connecting two stations of the network has a length, and the topological approximation, which neglects such lengths, is a poor description of the system. Of course, a generalization of the analysis to weighted networks would also extend the application of the small-world concept to a realm of new networks. A very significative example is that of transportation system: public transportation (bus, subway and trains), highways, airplane connections.

In [74] the authors present a way to extend the small-world analysis from topological to weighted networks. A weighted network can be characterized by introducing the variable *efficiency* E, which measures how efficiently the nodes exchange information. The definition of small-world behavior can be formulated in terms of the efficiency: this single measure evaluated on a global and on a local scale plays in turn the role of L and C. Small-world networks result as systems that are both globally and locally efficient [72]. The formalism is valid both for weighted and unweighted (topological) networks. In the case of topological networks, the measures introduced in [74] do not coincide exactly with those given by Watts and Strogatz. For example, the measures introduced in [74] also work in the case of unconnected graphs. An important quantity, previously not considered, is the *cost* of a network. Often high (global and local) efficiency implies a high cost of the network.

A weighted and possibly even non-connected and non-sparse graph is considered. A weighted graph needs two matrices to be described:

- The *adjacency matrix* $[a_{ij}]$, containing the information about the existence or not existence of a link, and defined as for the topological graph

as a set of numbers $a_{ij} = 1$ when there is an edge joining i to j, and $a_{ij} = 0$ otherwise;

- A matrix of the weights associated with each link. This matrix $[w_{ij}]$ is named the *matrix of physical distances* because the number w_{ij} can be imagined as the space distance between i and j. Moreover, w_{ij} is assumed to be known even if in the graph there is no edge between i and j.

To make a few concrete examples: w_{ij} can be identified with the geographical distance between stations i and j both in the case of the electrical power grid of the western United States studied by Watts and Strogatz and in the case of other transportation systems considered in [74]. In such a situation w_{ij} satisfy the triangular inequality, though in general this is not a necessary assumption. The presence of multiple edges, typical of the neural network of the C. elegans and of social systems like the network of film actors, can be included in the same framework by setting w_{ij} equal to the minimum between 1 and the inverse of the number of edges between i and j (respectively, the inverse of the number of junctions between two neurons or the inverse of the number of movies two actors did together). This allows to remove the hypothesis of simple network and to consider also non-simple systems as weighted networks. The resulting weighted network is, of course, a case in which the triangular inequality is not satisfied. For a computer network or Internet, w_{ij} can be assumed to be proportional to the time needed to exchange a unitary packet of information between i and j through a direct link. Or as $1/v_{ij}$, the inverse of the velocity of a chemical reaction along a direct connection in a metabolic network. Of course, in the particular case of an unweighted (topological) graph $w_{ij} = 1 \; \forall \, i \neq j$.

In a weighted graph the definition of the shortest path length d_{ij} between two generic points i and j is different from the definition used for an unweighted graph. In this case the shortest path length d_{ij} is in fact defined as the smallest sum of the physical distances throughout all the possible paths in the graph

from i to j. Again, when $w_{ij} = 1 \; \forall \, i \neq j$, i.e., in the particular case of an unweighted graph, d_{ij} reduces to the minimum number of edges traversed to get from i to j. The matrix of the shortest path lengths $[d_{ij}]$ is therefore calculated by using the information contained both in matrix $[a_{ij}]$ and in matrix $[w_{ij}]$ (for example, by using Dijkstra's algorithm ($O(N^2 \log N)$). It is supposed that every vertex sends information along the network through its edges and that the efficiency ϵ_{ij} in the communication between vertices i and j is inversely proportional to the shortest distance: $\epsilon_{ij} = 1/d_{ij} \; \forall \, i, j$. This is a reasonable approximation in general, but sometimes other relationships might be used, especially when justified by a more specific knowledge about the system. By assuming $\epsilon_{ij} = 1/d_{ij}$, when there is no path in the graph between i and j the shortest path length between them is $d_{ij} = \infty$ and consistently $\epsilon_{ij} = 0$. Consequently, the average *efficiency* of the graph can be defined as [112]

$$E = \frac{\sum_{i \neq j} \epsilon_{ij}}{N(N-1)} = \frac{1}{N(N-1)} \sum_{i \neq j} \frac{1}{d_{ij}} \, . \tag{1.4}$$

If the graph is undirected, i.e., there is no associated direction with the links, both $[w_{ij}]$ and $[d_{ij}]$ are symmetric matrices and therefore the quantity E can be defined simply by using only half of the matrix as

$$E = \frac{2}{N(N-1)} \sum_{i<j} \frac{1}{d_{ij}} \, .$$

Anyway the more general definition (1.4) allows to easily apply the presented formalism to directed graphs as well.

Formula (1.4) gives a value of E that can vary in the range $[0, \infty]$. It would be more practical to have E normalized to be in the interval $[0, 1]$. E can be normalized by considering the ideal case in which the graph has all the $N(N-1)/2$ possible edges. In such a case the information is propagated in the most efficient way and E assumes its maximum value. The efficiency E considered in the following is always normalized: $0 \leq E \leq 1$. Though the maximum value $E = 1$ is typically reached only when there is an edge between

each couple of vertices (in networks in which the triangular inequality holds), real networks can nevertheless assume high values of E.

One of the advantages of the efficiency-based formalism is that a single measure, the efficiency E (instead of the two different measures L and C used in the Watts Strogatz (WS) formalism) is sufficient to define the small-world behavior. In fact, on one hand, the quantity defined in (1.4) can be evaluated as it is for the whole graph to characterize the *global efficiency* of the graph; in this case, it is denoted by E_{glob}. Being the efficiency in communication between two generic vertices, E_{glob} plays a role similar to the inverse of the characteristic path length L. In fact, L is the mean of d_{ij}, while E_{glob} is the average of $1/d_{ij}$, i.e., the inverse of the harmonic mean of $[d_{ij}]$. Nowadays the harmonic mean finds extensive applications in a variety of different fields: in particular it is used to calculate the average performance of computer systems [86, 112], parallel processors [57] and communication devices (for example, modems and Ethernets [59]). In all such cases, where a mean flow-rate of information has to be computed, the simple arithmetic mean gives the wrong result. In some cases $1/L$ gives a good approximation of E_{glob}, although E_{glob} is the real variable to be considered to characterize the efficiency of a system transporting information in parallel. In the particular case of a disconnected graph the difference between the two quantities is evident because $L = \infty$ while E_{glob} is a finite number.

On the other hand the same measure, the efficiency, can be evaluated for any subgraph of the graph and therefore it can be used also to characterize the local properties of the graph itself. In the WS formalism it is not possible to use the characteristic path length for quantifying both the global and the local properties of the graph simply because L cannot be calculated locally, most of the subgraphs of the neighbors of a generic vertex i being disconnected. Since E is defined also for a disconnected graph, it is possible to characterize the local properties of the graph by evaluating for each vertex i the efficiency of the subgraph of the neighbors of i. The *local efficiency* E_{loc} is defined as

the average of the N values obtained in this way. Here, for each vertex i, the normalization factor is the the efficiency of the ideal case in which the subgraph has all the $k_i(k_i-1)/2$ possible edges, where k_i is the number of neighbors of vertex i. E_{loc} plays a role similar to the clustering coefficient C: since vertex i does not belong to the relative subgraph, the local efficiency E_{loc} tells how much the system is fault tolerant, thus how efficient is the communication between the first neighbors of i when i is removed. This concept of fault tolerance is different from that adopted in [11, 29, 33], where the authors consider the response of the entire network to the removal of a node i. Here the response of the subgraph of first neighbors of i to the removal of i is considered.

A new, generalizing, definition of small world can be introduced, built in terms of the characteristics of information flow at global and local level: a small-world network is a network with high E_{glob} and E_{loc}, i.e., very efficient both in global and local communication [74]. This definition is valid both for unweighted and for weighted graphs, and can also be applied to disconnected graphs and/or non-sparse graphs.

1.3.4 Comparison between E_{glob}, E_{loc} and L, C

In [74] the authors also study the correspondence between their measures and the quantities L and C defined in [126] (or, correspondingly, $1/L$ and C). The fundamental difference is that $1/L$ measures the efficiency of a sequential system, that is to say, of a system where there is only one packet of information going along the network. On the other hand, E_{glob} measures the efficiency for parallel systems, where all the nodes in the network concurrently exchange packets of information. This can explain why L works reasonably: it can be seen that $1/L$ is a reasonable approximation of E_{glob} when there are not huge differences among the distances in the graph, and so considering just one packet in the system is more or less equivalent to the case where multiple packets are present. This is the case for all the networks presented in [126],

and this effect is strengthened even more by the fact that the topology only is considered.

Having explained why L behaves relatively well in some case, it is also worth noticing that, like every approximation, it fails to properly deal with all cases. For example, the sequentiality of the measure $1/L$ explains why many limitations have to be introduced, like connectedness, that are present just in order to make the formulas valid. Consider the limit case where a node is isolated from the system. In the case of a neural network, this corresponds for example to the death of a neuron. In this case, $1/L$ drops to zero ($L = \infty$), which is of course not the overall efficiency of the system: in fact, the brain continues to work, as all the other neurons continue to exchange information; only, the efficiency is just slightly diminished, as now there is one neuron less and, correctly, this is properly taken into account using E_{glob}.

Even without dropping the connectedness assumption, another example can show how in the limit case, the approximation given by $1/L$ diverges from the real efficiency measure. Fig. 1.11 shows the addition of a new node to the Internet (which already had N nodes), with efficiency ε, that can be seen as the speed of the connection. This happens every time the Internet is augmented with a new computer and every time a computer is turned on. A situation like this occurs daily in the order of the millions. How does it globally affect the Internet, according to L and E_{glob}? It can be proved that L augments by approximately $\frac{1}{\varepsilon(N+1)}$. This means that if for any reason, the connection speed is particularly slow (or becomes such, for example due to a congestion, or the computer gets low in resources), L of the whole Internet is heavily affected and can rapidly become enormous. Even, whenever the computer blocks (or it is shut down), L diverges to infinity (like, so to say, if the Internet had collapsed). On the other hand, the efficiency E_{glob} has a relative decrement of approximately $\frac{2}{N+1}$, which means that in practice, as N is quite large, the particular behavior of the new computer affects the Internet in a negligible way. Summing up, having one or few computers with an extremely

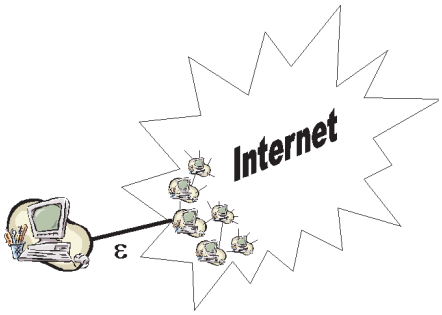

Figure 1.11: A new computer is attached to the Internet (which already had N nodes) with a connection represented by a small efficiency ε. Having one (or few) computer with an extremely slow connection, does not mean that the whole Internet diminishes by far its efficiency: in practice, the presence of such slow computer goes unnoticed, because the other thousands of computers are exchanging packets among them in a very efficient way. L fails to properly capture the global behavior of systems like the Internet, unlike E_{glob}, that perfectly matches the observed behavior [74].

slow connection, does not mean that the whole Internet diminishes by far its efficiency: in practice, the presence of such few very slow computers goes unnoticed, because the other thousands of computers are exchanging packets among them in a very efficient way. Therefore, L fails to properly capture the global behavior of systems like the Internet ($1/L$ would give a number very close to zero, because it measures the average efficiency in case a single packet is active through the Internet), unlike E_{glob}, that perfectly matches the observed behavior.

The crucial point here is the following: all the networks considered in [126] to justify the definition of small worlds (and, in fact, most of the networks that model complex systems) are parallel systems, where all the nodes interact in parallel (Internet, World Wide Web, social networks, neural systems and so on). With this assumption, E_{glob} measures the real efficiency of the system, and $1/L$ is just a first rough approximation, as it deals with the sequential case only. As for C and E_{loc}, it can be shown that C, in the case of undirected topological graphs, is always a reasonable approximation of E_{loc}. Therefore, the seemingly ad hoc nature of C in the WS formalism now finds a new meaning in the general notion of efficiency. There are not two different kinds of properties to consider when analyzing a network on the local and on the global scale, but just one unifying concept: the efficiency to transport information [74].

1.3.5 The cost of a network

An important variable to take into account, especially when weighted networks are considered and when different real systems have to be analyzed and compared, is the cost of a network. In fact, the efficiency of a graph is expected to be higher as the number of edges in the graph increases. As a counterpart, in any real network there is a price to pay for number and length (weight) of edges. In particular the 'short cuts', i.e., the rewired edges that produce the rapid drop of L and the onset of the small-world behavior in the WS model

(Sec. 1.3.2), connect at no cost vertices that would otherwise be much farther apart.

It is therefore crucial to consider weighted networks and to define a variable to quantify the cost of a network. In order to do so, in [74] the authors define the *cost* of a graph as:

$$Cost = \frac{\sum_{i \neq j} a_{ij}\, \gamma(w_{ij})}{\sum_{i \neq j} \gamma(w_{ij})}\,. \tag{1.5}$$

Here, γ is the so-called cost evaluator function, which calculates the cost needed to build up a connection with a given length. Of course, γ could be equivalently defined on efficiencies rather than distances (so, indicating in a sense the cost to set up a communication channel with the given efficiency). The cost of the ideal graph (in which all the possible edges are present) is already included in the denominator of this definition. Because of such a normalization, the γ function needs only to be defined up to a multiplicative constant, and the quantity $Cost$ is defined in the interval $[0, 1]$, assuming the maximum value 1 for the ideal graph, i.e., when all the edges are present in the graph. Denoting by K the number of edges in the graph, in the case of an unweighted graph (for example, the WS model) $Cost$ reduces to the normalized number of edges $2K/(N(N-1))$.

Unless otherwise specified, in the following it is assumed that γ is defined as the identity function: $\gamma(x) = x$. In fact, such a cost evaluator works for unweighted networks and also for most of the real networks, those where the cost of a connection is proportional to its length (to the Euclidean distance, for example): in all such cases the definition of the cost reduces to $Cost = (\sum_{i \neq j} a_{ij} w_{ij})/(\sum_{i \neq j} w_{ij})$. A different definition of the cost evaluator function will be used instead when networks with multiple edges are represented as weighted graphs (for example in the weighted C. elegans and in the weighted movie actors).

With the formalism based on the two efficiencies E_{glob} and E_{loc}, and on the variable $Cost$, all defined in the range from 0 to 1, it is possible to study

in an unified way unweighted (topological) and weighted networks. Therefore, the following key notion can be defined: a network is called economic if it has a low $Cost$; then, an economic small world is a network having high E_{glob} and E_{loc}, and low $Cost$ (i.e., both economic and small-world) [74].

1.3.6 The economic small-world behavior

It is now possible to illustrate the three quantities E_{glob}, E_{loc} and $Cost$ at work in some practical examples. Starting from the original WS model (Sec. 1.3.2), and proceeding with different models, it will be shown how these three quantities behave in a dynamic environment where the network changes, have some nontrivial interaction among each other, and give birth to small worlds [74].

Throughout the models and all the real networks presented here and in the next section, the matrix $[d_{ij}]$ is computed by using two different methods: the Floyd-Warshall algorithm ($O(N^3)$) [51] and the Dijkstra's algorithm ($O(N^2 \log(N))$ [88].

Model 1 (the WS model) is a procedure to construct a family of unweighted networks with a fixed cost. Model 2 is a way to construct unweighted networks, this time with increasing cost. Model 3 and model 4 are two examples of weighted networks. In particular in model 4 the length of the edge connecting two nodes is the Euclidean distance between the nodes.

Model 1) The original WS model is unweighted (topological): this means it is possible to set $w_{ij} = 1 \ \forall\, i \neq j$, and the quantities d_{ij} reduce to the minimum number of edges to get from i to j. The dynamic changes of the network consist in rewirings: since the weight is the same for all edges, also for rewired edges, this means that the $Cost$ (that is proportional to the total number of edges K) does not change with the rewiring probability p.

Fig. 1.12 refers to a regular lattice with $N = 1000$ and three different values of k ($k = 6, 10, 20$), corresponding to networks with different (low) cost (respectively $Cost = 0.006, 0.01, 0.02$); the efficiencies E_{glob} and E_{loc} are reported as functions of p. For $p = 0$ the system is expected to be inefficient on

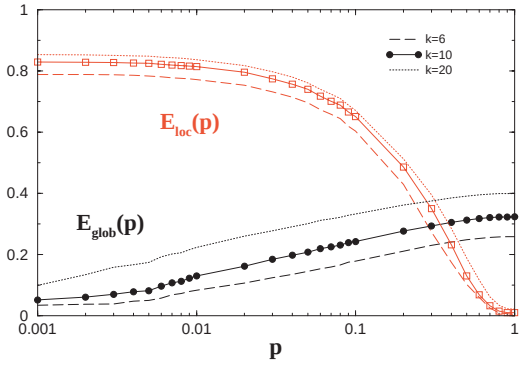

Figure 1.12: Global and local efficiency for model 1 (the WS model), the class of topological graphs considered by Watts and Strogatz. A regular lattice with $N = 1000$ and k edges per node is rewired with probability p. The logarithmic horizontal scale is used to resolve the rapid increase in E_{glob} due to the presence of short cuts and corresponding to the onset of the small world. During this increase, E_{loc} remains large and almost equal to the value for the regular lattice. Small worlds have high E_{glob} and E_{loc}. Three different values $k = 6, 10, 20$ corresponding respectively to $Cost = 0.006, 0.01, 0.02$ are considered. Here and in the following figures the efficiency and the cost are dimensionless quantities normalized to the values of the ideal graph [74].

a global scale (an analytical estimate gives $E_{glob} \sim k/N \log(N/K)$), but locally efficient. The situation is inverted for random graphs. In fact, for example in the case $k = 20$, at $p = 1$ E_{glob} assumes a maximum value of 0.4, meaning 40% of the efficiency of the ideal graph with an edge between each couple of vertices. This happens at the expenses of the fault tolerance ($E_{loc} \sim 0$). The (economic) small-world behavior appears for intermediate values of p. It results from the fast increase of E_{glob} caused by the introduction of only a few rewired edges (short cuts), which on the other hand do not affect E_{loc}. For the case $k = 20$, at $p \sim 0.1$ E_{glob} has almost reached the maximum value of 0.4, though E_{loc} has only diminished by very little from the maximum value of 0.82.

For such an unweighted case the description in terms of network efficiency is similar to that given by Watts and Strogatz. In Fig. 1.13 it is shown that if

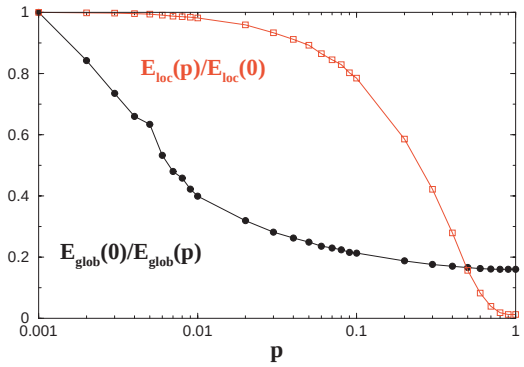

Figure 1.13: Model 1 (the WS model). A regular lattice with $N = 1000$ and $k = 10$ edges per node is rewired with probability p. Reporting the quantities $(\frac{E_{glob}(p)}{E_{glob}(0)})^{-1}$ and $\frac{E_{loc}(p)}{E_{loc}(0)}$ as functions of p, the two curves show a behavior similar respectively to $\frac{L(p)}{L(0)}$ and $\frac{C(p)}{C(0)}$ [74].

the quantities $1/E_{glob}(p)$ and $E_{loc}(p)$ are reported, and a normalization similar to that adopted by Watts and Strogatz is used, i.e., $E_{glob}(0)/E_{glob}(p)$ and $E_{loc}(p)/E_{loc}(0)$, curves with qualitatively the same behavior of the curves $L(p)/L(0)$ and $C(p)/C(0)$ are obtained (compare with Fig. 1.9).

Model 2) The above model has proved successful in order to produce small worlds, i.e., networks with high E_{glob} and high E_{loc}. However, if that is the goal, then there are much simpler procedures that can output a small world, even starting from an arbitrary configuration. For example, Fig. 1.14 refers to a model where, starting from a configuration with $N = 100$ nodes and

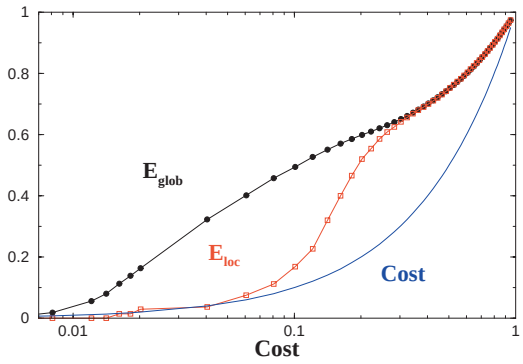

Figure 1.14: Model 2. A network is created by adding links randomly to an initial configuration with $N = 100$ nodes and no links. E_{glob} and E_{loc} are plotted as functions of the $Cost$. The identity curve $Cost$ is also reported to help the reader since a logarithmic horizontal scale is used [74].

no links, new links are added randomly, until a completely connected network is obtained. This model is unweighted as model 1. Contrarily to the case of model 1, the network changes by adding links, then the cost is not a fixed quantity but varies in a monotonic way, increasing every time a new link is added. For $Cost \sim 0.5 - 0.6$, a small-world network with $E_{glob} = E_{loc} = 0.8$ is obtained. So, if this trivial method manages to produce small worlds, why is it so hard to find many small worlds like these in nature? The obvious answer is that here, a small world is obtained at the expense of the cost: with rich resources (high cost), the small-world behavior always appears. In fact, in the limit of the completely connected network ($Cost = 1$), $E_{glob} = E_{loc} = 1$. But what also matters in nature is economy of a network, and in fact a trivial technique like this fails to produce economic small worlds.

The relationship of the variable cost with respect to the other two variables is not that trivial. Even in the very simple and rigid "monotonic" setting dictated by this model, an interesting behavior of the variables E_{glob} and E_{loc} as functions of $Cost$ can be observed. In particular, E_{loc} rapidly rises when the cost increases from 0.1 to 0.2. This means that moving from $Cost = 0.1$ to $Cost = 0.2$, the local efficiency of the network can be increased from $E_{loc} = 0.1$ to $E_{loc} = 0.6$. A network with 60% of the efficiency of the ideal network both on a global and local scale is therefore obtained, with only 20% of the cost: this is an example of an economic small-world network. The observed effect has an higher probability to happen in the mid-area in between the areas of low cost and high cost, and it is a first sign that complex interactions do occur, but not with very low cost or with very high cost (where economic small worlds cannot be found).

Model 3) In this third model, features of the previous models 1 and 2 are combined: rewiring as in model 1, monotonic increase of the cost as in model 2. So, while in model 1 the short cuts connect at no cost (because $w_{ij} = 1 \; \forall \, i \neq j$) vertices that would otherwise be much farther apart (which is a rather unrealistic assumption for real networks), in this model each rewiring has a cost.

In Fig. 1.15 a random rewiring, in which the length of each rewired edge is set to change from 1 to 3, is implemented. Note that this model, unlike the previous two, is weighted. The figure shows that the small-world behavior is still present even when the length of the rewired edges is larger than the original one. For p around the value 0.1, E_{glob} has almost reached the maximum value 0.18 (18% of the global efficiency of the ideal graph with all couples of nodes directly connected with edges of length equal to 1) while E_{loc} has not changed too much from the maximum value 0.8 (assumed at $p = 0$). The only difference with respect to model 1 is that the behavior of E_{glob} is not simply monotonic increasing. Of course in this model the variable $Cost$ increases with p. It is interesting to notice that the curve $Cost$ as a function of p, plotted in

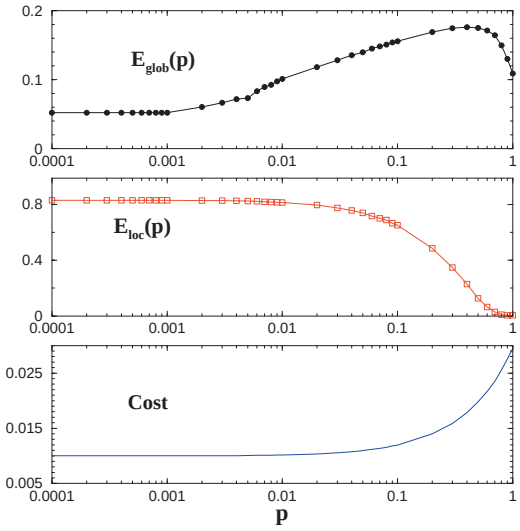

Figure 1.15: The three quantities E_{glob}, E_{loc} and $Cost$ are reported as functions of p in model 3. Starting with a regular lattice with $N = 1000$ and $k = 10$, the same rewiring procedure as in the WS model is implemented, with the only difference that the length of the rewired edge is set to change from the value 1 to the value 3. The economic small-world behavior shows up for $p \sim 0.1$ [74].

the bottom of the figure, is specular to the curve E_{loc} as a function of p. This means that in the small-world situation, the network is also economic, in fact the *Cost* stays very close to the minimum possible value (assumed of course in the regular case $p = 0$).

The robustness of the obtained results has been checked by increasing even more the length of the rewired edges. Therefore, this model shows that to some extent, the structure of a network plays a relevant role in the economy. Also, note that in this more complex (weighted) model, the behaviors of E_{loc} and E_{glob} become more complex as well: now, E_{glob} is not a monotonic function of the cost any more, and E_{loc} is monotonic, but decreasing. So, the introduction of the weighted model further shows how the relative behavior of the three variables E_{glob}, E_{loc} and $Cost$ is far from simple.

Model 4) This example is builded on model 3, grounding it more in reality using a real geometry, in order to further investigate whether the above effects can also appear in real networks which are not just mathematical possibilities. In this weighted model, the length of the edge connecting two nodes is the Euclidean distance between the nodes. The nodes can be placed with different geometries. Here the case in which the N nodes are placed on a circle, as in the WS model, is considered. Now the geometry is important because the physical distance between nodes i and j $(i, j = 1, \ldots, N)$ is defined as the Euclidean distance between i and j. In the case of nodes on a circle,

$$w_{ij} = \frac{\sin(|i-j|\pi/N)}{\sin(\pi/N)}\,. \tag{1.6}$$

In this formula, the length of the edge between two neighbors is set to be equal to 1, i.e., $w_{ij} = 1$ when $|i - j| = 1$. The radius of the circle is then $R = 1/2\ \sin(\pi/2)/\sin(\pi/N)$.

Fig. 1.16 shows the results obtained by implementing a rewiring procedure similar to that considered in the previous models. The only difference with respect to the previous case is that now it is not possible to start from a lattice with $N = 1000$, $k = 10$. Such a network, in fact, when considered with the metrics in (1.6) would have $K = 5000$ edges and a too high global

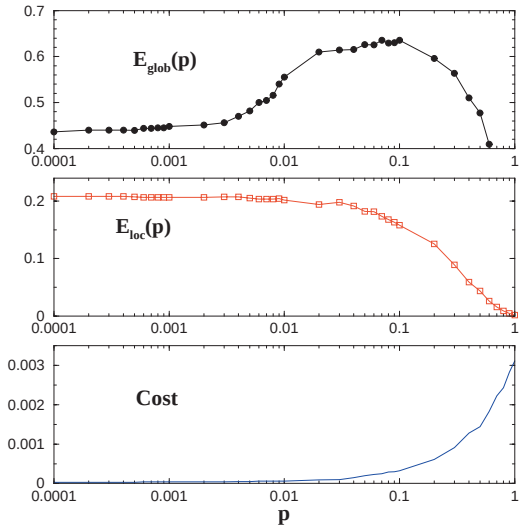

Figure 1.16: The three quantities E_{glob}, E_{loc} and $Cost$ are reported as functions of p in model 4. Starting with a regular lattice with $N = 1000$ and a total number of edges $K = 1507$, the rewiring procedure with probability p is implemented. The economic small-world behavior shows up for $p \sim 0.02 - 0.04$ [74].

efficiency, about 99% of the ideal graph. On the other hand, considering as a starting network a lattice with $k = 2$ would affect the local efficiency. Then the procedure is as follows. A regular network with $N = 1000$ and $k = 6$ is created and then 50% of the 3000 edges are randomly eliminated to decrease the global efficiency: in the random realization reported in figure $K = 1507$ edges are left. At this point, the usual rewiring process can be implemented on this network. For $p \sim 0.02 - 0.04$, E_{glob} has almost reached its maximum value 0.62 while E_{loc} has not changed much from the maximum value 0.2 (assumed at $p = 0$). As in model 3, the behavior of E_{glob} is not simply monotonic decreasing and as in model 3 the small-world network is also an economic network, i.e., the $Cost$ stays very close to the minimum possible value (assumed of course for $p = 0$).

So, this model and model 3 suggest that the economic small-world behavior is not only an effect of the topological abstraction but can also be found in all the weighted networks where the physical distance is important and the rewiring has a cost (and show how intricate the relative behavior of E_{glob}, E_{loc} and $Cost$ can be).

1.3.7 Efficiency and cost of real-world networks

With the formalism based on the three quantities E_{glob}, E_{loc} and $Cost$, all defined in the range from 0 to 1, it is possible to study in a unified way unweighted (topological) and weighted networks, also considering some empirical examples. In [74] the authors present a study of:

1. Neural networks (two examples of networks of cortico-cortical connections and an example of a nervous system at the level of connections between neurons);
2. Social networks (the collaboration network of movie actors);
3. Communication networks (the World Wide Web and the Internet);
4. Transportation systems (the Boston urban transportation systems).

Neural networks

The brain is the most complex and fascinating information transportation system. Its staggering complexity is the evolutionary result of adaptivity, functionality and economy. The brain complexity is already reflected in the complexity of its structure [69]. Of course neural structures can be studied at several levels of scale. In fact, thanks to recent experiments, a wealth of neuroanatomical data ranging from the fine structure of connectivity between single neurons to pathways linking different areas of the cerebral cortex is now available. The authors focus first on the analysis of the neuroanatomical structure of cerebral cortex and then on a simple nervous system at the level of wirings between neurons.

1) Networks of cortico-cortical connections. The anatomical connections between cortical areas and groups of cortical neurons are of particular importance because they are considered to have an intricate relationship with the functional connectivity of the cerebral cortex [114].

Two databases of cortico-cortical connections in the macaque and in the cat are analyzed [105]. The databases consist of the wiring diagrams of the two systems and there is no information about the weights associated with the links: therefore, these systems will be studied as unweighted networks. The macaque database contains $N = 69$ cortical areas and $K = 413$ connections ([130], cortical parcellation after [50], except auditory areas which follow [97]). The cat database has instead $N = 55$ cortical areas (including hippocampus, amygdala, entorhinal cortex and subiculum) and $K = 564$ connections (revised database and cortical parcellation from [106]).

The results in the first two lines of Table 1.2 indicate that the two networks are economic small worlds: they have high global efficiency (respectively 52% and 69% of the efficiency of the ideal graph) and high local efficiency (70% and 83% of the ideal graph), i.e., high fault tolerance [110] with only 18% and 38% of the wirings. Moreover, E_{glob} is similar to the value for random graphs, while E_{loc} is larger than E_{loc}^{random}. These results indicate that in neural

Table 1.2: The macaque and cat cortico-cortical connections [105] are two unweighted networks with respectively $N = 69$ and $N = 55$ nodes, $K = 413$ and $K = 564$ connections. Global efficiency, local efficiency and cost are reported in the first two lines of the table. The results are compared with the efficiency of random graphs. The random graphs used as a null model are created by distributing the K connections between randomly chosen couples of nodes [23]. The estimated E_{glob}^{random} and E_{loc}^{random} are averages over several realizations. The nervous system of C. elegans is better described by a weighted network: the network consists of $N = 282$ nodes and $K = 2462$ edges which can be of two different kinds, either synaptic connections or gap junctions. In the third line of the table the result for the C. elegans considered as unweighted (to compare with cortico-cortical networks) is reported, while in the fourth line the weights are considered. All these systems are examples of economic small worlds [74].

Unweighted:	E_{glob}	E_{glob}^{random}	E_{loc}	E_{loc}^{random}	$Cost$
Macaque	0.52	0.57	0.70	0.35	0.18
Cat	0.69	0.69	0.83	0.67	0.38
C. elegans	0.46	0.48	0.47	0.12	0.06

Weighted:	E_{glob}	E_{loc}	$Cost$
C. elegans	0.35	0.34	0.18

cortex each region is intermingled with the others and has grown following a perfect balance between cost, local necessities (fault tolerance) and wide-scope interactions.

2) A network of connections between neurons. As a second example, the neural network of C. elegans is considered; it is the only case of a nervous system completely mapped at the level of neurons and chemical synapses [127]. The database studied here is the same considered by Watts and Strogatz and is taken from [1].

As already discussed in Sec. 1.3.3, the nervous system of C. elegans is better described by a weighted network. In fact, the C. elegans is a multiple edges system, i.e., there can be more than one edge (up to 72 edges) between the same couple of nodes i and j. The presence of multiple edges can be expressed in the weighted networks formalism described in Sec. 1.3.3 by considering a simple but weighted graph, and setting w_{ij} equal to the inverse of the number of edges between i and j. In this way, a weighted network consisting of $N = 282$ nodes and $K = 2462$ edges is obtained (an edge $i - j$ is defined by the presence of at least one synaptic connection or gap junction). Doing this choice to weight the system implies to define appropriately the cost evaluator function γ (which cannot be the identity any more): the correct choice is to set $\gamma(x) = 1/x$, that is to say, the cost of a connection is the number of synaptic connections and gap junctions that make it.

In order to compare the C. elegans with the two cortico-cortical connections networks, it is first considered as an unweighted network neglecting the information contained in $[w_{ij}]$ (as if $w_{ij} = 1 \; \forall \, i \neq j$). Similarly to the two cortico-cortical connections networks, the unweighted C. elegans is also an economic small-world network. The third line of Table 1.2 shows that with a relative low cost (6% of the wirings), C. elegans achieves about 50% of both the global and local efficiency of the ideal graph (see also the comparison with the random graph). Moreover, the value of E_{glob} is similar to E_{loc}. This is a difference from cortex databases, where fault tolerance is slightly privileged

with respect to global communication.

Finally, the C. elegans can be considered in all its completeness, i.e., as a weighted graph. Of course in this case the random graph does not give any more the best approximation for E_{glob}. Nevertheless the values of E_{glob}, E_{loc} and $Cost$ have a meaning by themselves, being normalized to the case of the ideal graph. The fourth line of Table 1.2 shows that the C. elegans is also an economic small world when considered as a weighted network with about 35% of the global and local efficiency of the ideal graph, obtained with a cost of 18%. It is interesting to notice that, as in the unweighted case, the system has similar values of E_{glob} and E_{loc} (that is, it behaves globally in the same way as it behaves locally).

The connectivity structure of the three neural networks studied reflects a long evolutionary process driven by the need to maximize global efficiency and to develop a robust response to defect failure (fault tolerance). All this at a relatively low cost, i.e., with a small number of edges, or with a minimum amount of the length of the wirings.

Social networks

As an example of social networks, in [74] the authors study the collaboration network of movie actors extracted from the Internet Movie Database, as of July 1999. The graph considered has $N = 277\,336$ nodes and $K = 8\,721\,428$ edges, and is not a connected graph.

The approach of Watts and Strogatz cannot be applied directly and they have to restrict their analysis to the giant connected component of the graph [126]. Here the authors apply their small-world analysis directly to the whole graph, without any restriction. Moreover, the unweighted case only provides whether actors participated in some movie together, or if they did not at all.

Of course, in reality there are instead various degrees of correlation: two actors that have done ten movies together are in a much stricter relation rather than two actors that have acted together only once. As in the case of the C.

elegans, this different degree of friendship can be better shaped by using a weighted network: the distance w_{ij} between two actors i and j is set as the inverse of the number of movies they did together.

As in the case of the C. elegans, together with this choice to weight the system, the cost evaluator function γ has to be defined appropriately: the correct choice is (again) to set $\gamma(x) = 1/x$, that is to say, the cost of a connection between two persons is the number of movies they did together.

The numerical values in Table 1.3 indicate that both the unweighted and

Table 1.3: The collaboration network of movie actors (extracted from the Internet Movie Database, `http://www.imdb.com`) can be described by an unweighted or a weighted graph with $N = 277\,336$ nodes and $K = 8\,721\,428$ edges [74].

Unweighted:	E_{glob}	E_{glob}^{random}	E_{loc}	E_{loc}^{random}	$Cost$
Movie actors	0.37	0.41	0.67	0.00026	0.0002

Weighted:	E_{glob}	E_{loc}	$Cost$
Movie actors	0.29	0.52	0.0005

the weighted network show the economic small-world phenomenon. In both cases, cost comes out as a leading principle: this is due somehow to physical limitations, as it is not easy for actors to perform in a huge number of movies, and for most of them their career is in any case limited in time, while the database spans all the temporal age.

Of course other social systems can be studied by means of this formalism: for example, the collaboration network of physicists [87, 88], the collaboration network of Marvel comics characters [8], or some other databases of social communities [14, 52].

Communication networks

Communication networks are ubiquitous nowadays: the so-called "information society" heavily relies on such networks to rapidly exchange information in a distributed fashion, all over the world.

In [74] the authors consider the two most important large-scale communication networks present nowadays: the World Wide Web and the Internet. Note that despite these two networks are often confused and identified, they are fundamentally different: the World Wide Web (WWW) network is based on information abstraction, via the fundamental concept of URI (Uniform Resource Identifier); so, it is not a physical structure, but an abstract structure. On the other hand, the Internet is a physical communication network, where each link and node have a physical representation in space. So, despite these two communication networks share lot of commonalities (last but not least, the fact the WWW essentially relies on the Internet structure to work), they are bottom-down deeply different: one network (WWW) is purely conceptual, the other one (the Internet) is physical.

The authors have studied a database of the World Wide Web with $N = 325\,729$ documents and $K = 1\,090\,108$ links, and a network of Internet with $N = 6474$ nodes and $K = 12\,572$ links. Both networks are considered as unweighted graphs.

Table 1.4 reports the result of the efficiency-cost analysis of the two networks. They have relatively high values of E_{glob} (slightly smaller than the best possible values obtained for random graphs) and E_{loc}, together with a very small cost: therefore, both of them are economic small worlds.

Observe that interestingly, despite the WWW is a virtual network and the Internet is a physical network, at a global scale they transport information essentially in the same way (as their E_{glob}'s are almost equal). At a local scale, the larger E_{loc} in the WWW case can be explained both by the tendency in the WWW to create Web communities (where pages talking about the same subject tend to link to each other), and by the fact that many pages within

Table 1.4: Communication networks. Data on the World Wide Web from `http://www.nd.edu/~networks` contains $N = 325\,729$ documents and $K = 1\,090\,108$ links [10], while the Internet database is taken from `http://moat.nlanr.net` and has $N = 6474$ nodes and $K = 12\,572$ links. Both systems are studied as unweighted graphs and are examples of economic small worlds [74].

	E_{glob}	E_{glob}^{random}	E_{loc}	E_{loc}^{random}	$Cost$
WWW	0.28	0.28	0.36	0.000001	0.00002
Internet	0.29	0.30	0.26	0.0005	0.006

the same site are often quickly connected to each other by some root or menu page.

As far as the cost is concerned, it is striking to notice how economic these networks are (for example, compare these data with the corresponding ones for the cases of neural networks). This clearly indicates that economy is a fundamental construction principle of the Internet and of the WWW.

Transportation networks

Another example of man-made networks are the transportation networks. As a paradigmatic example of a system belonging to this class, in [74] the authors consider the Boston public transportation system. Another transportation network is the network of airplane and highway connections throughout the world.

The Boston subway transportation system (MBTA) is the oldest subway system in the US (the first electric streetcar line in Boston, which is now part of the MBTA Green Line, began operation on January 1, 1889) and consists of $N = 124$ stations and $K = 124$ tunnels (connecting couples of stations) extending throughout Boston and the other cities of the Massachusetts Bay. As some of the previous databases, this is another example of a network bet-

ter described by a weighted graph: in this case the matrix $[w_{ij}]$ is given by the Euclidean distance between i and j, i.e., by the geographical distances between stations. In this sense the MBTA is a weighted network more similar to the electrical power grid of the western United States than to weighted networks representing multiple edges systems like the neural network of the C. elegans or the network of film actors. In fact, in the case of the MBTA the quantities w_{ij} satisfy the triangle inequality and the definition of the ideal graph is straightforward since the spatial distance w_{ij} between stations i and j is perfectly defined, independently from the existence or not of the edge $i - j$. In particular the matrix $[w_{ij}]$ has been calculated by using information databases from the MBTA, from the Geographic Data Technology (GDT) and the US National Mapping Division. The authors first study the system in the unweighted approximation, illustrating how in this case L and C do not work, and they must use the efficiency-based formalism; they finally study the efficiency of the MBTA in its completeness, as a weighted network [72, 73].

In the unweighted network approximation the information contained in $[w_{ij}]$ is not used (as if $w_{ij} = 1 \ \forall i \neq j$). Now, consider for example L: applying to the MBTA the WS formalism presented in Sec. 1.3.2, valid for unweighted (topological) networks, gives $L = 15.55$ (an average of 15 steps, or 15 stations to connect 2 generic stations). And now, to decide if the MBTA is a small world the obtained L has to be compared with the respective values for a random graph with the same N and K. But considering random graphs gives rise to disconnected graphs and $L = \infty$. So, it is not possible to draw any conclusion. On the other hand, the same unweighted network can be perfectly studied by using the efficiency formalism of Sec. 1.3.3. The problem of the divergence of L is here avoided, because when there is no path in the graph between i and j, $d_{ij} = \infty$ and consistently $\epsilon_{ij} = 0$. The results are reported in the first line of Table 1.5 and compared with the values obtained for the random graph with same number of N and K (as said before, in the unweighted case, the random graph provides the best value of E_{glob}). It is

Table 1.5: The MBTA can be considered as a network of $N = 124$ nodes and $K = 124$ links. The MBTA is first studied as an unweighted network and then as a weighted network. Finally the weighted network consisting in the underground transportation system plus the bus transportation system is considered as a more complete transportation system. The matrix $[w_{ij}]$ has been calculated by using databases from the MBTA (`http://www.mbta.com`) and the US National Mapping Division [74].

Unweighted:	E_{glob}	E_{glob}^{random}	E_{loc}	E_{loc}^{random}	$Cost$
MBTA	0.10	0.14	0.006	0.015	0.016

Weighted:	E_{glob}	E_{loc}	$Cost$
MBTA	0.63	0.03	0.002
MBTA + bus	0.72	0.46	0.004

easy to see that the unweighted network is not a small world because the E_{loc} should be much larger than E_{loc}^{random} and is instead smaller than E_{loc}^{random}.

The second line of Table 1.5 reports the results for the weighted case, i.e., the case in which the link characteristics (lengths in this case) are properly taken into account, and not flattened into their topological abstraction. As a main difference from the unweighted case considered before, in a weighted case the random graph does not give the estimate of the highest global efficiency. In any case the quantities E_{glob} and E_{loc} have a meaning by themselves because of the adopted normalization: the numbers show MBTA is a very efficient transportation system on a global scale but not at the local level. In fact, $E_{glob} = 0.63$ means that MBTA is only 37% less efficient than the ideal subway with a direct tunnel from each station to the others. On the other hand, $E_{loc} = 0.03$ indicates a poor local efficiency: differently from a neural network or from a social system, the MBTA is not fault tolerant and a damage in a station will dramatically affect the efficiency in the connection between the previous and the next station.

To understand better the difference with respect to the other systems previously considered, few general considerations about the variable *Cost* and the rationales in the construction principles have to be made. As said before, in general the efficiency of a graph increases with the number of edges. As a counterpart, in any real network there is a price to pay for number and length (weight) of edges. The cost of the weighted MBTA is $Cost = 0.002$, a value much smaller than those obtained for example for the three neural networks considered, respectively $Cost = 0.18, 0.38, 0.06 - 0.07$. This means that MBTA achieves the 63% of the efficiency of the ideal subway with a cost of only the 0.2%. The price to pay for such low-cost high global efficiency is the lack of fault tolerance. The difference with respect to neural networks comes from different needs and priorities in the construction and evolution mechanism.

A neural network is the result of perfect balance between global and local efficiency. On the other hand, when a subway system has to be built, the priority is given to the achievement of global efficiency at a relatively low cost, and not to fault tolerance. In fact, a temporary problem in a station can be solved in an economic way by other means: for example, walking, or taking a bus from the previous to the next station. That is to say, the MBTA is not a closed system: it can be considered, after all, as a subgraph of a wider transportation network. This property is very often so understood that it is not even noted (consider, for example, the case of the brains), but it is nevertheless of fundamental importance for the analysis of a system: while global efficiency is without doubt the major characteristic, it is closure that somehow leads a system to have high local efficiency (without alternatives, there should be high fault tolerance).

The MBTA is not a closed system, and thus this explains why, unlike in the case of neural networks, fault tolerance is not a critical issue. Changing the MBTA network to take into account, for example, the bus system, indeed, this extended transportation system comes back to be an economic small-

world network. In fact, the numbers in the third line of Table 1.5 indicate that the extended transportation system achieves high global but also high local efficiency ($E_{glob} = 0.72$, $E_{loc} = 0.43$), at a still low price ($Cost$ has only increased from 0.002 to 0.004). Qualitatively similar results have been obtained for other underground systems.

Transportation systems can, of course, also be analyzed at different scales: a similar analysis on a wider transportation system, consisting of all the main airplane and highway connections throughout the world, shows a small-world behavior. This can be explained by the fact that in such a system almost all the reasonable transportation alternatives available at that scale are considered. In this way the system is closed, i.e., there are no other reasonable routing alternatives, and so fault tolerance comes back, after the cost, as a leading construction principle.

1.3.8 Degree distributions

As said in Sec. 1.1, the degree of a vertex in a network is the number of edges incident on (i.e., connected to) that vertex. Denoting by p_k the fraction of vertices in the network having degree k, i.e., the probability that a vertex chosen uniformly at random has degree k, the degree distribution for the network is the histogram of the degrees of vertices. In a random graph of the type studied by Erdős and Rényi [45, 46, 47], each edge is present or absent with equal probability, and hence the degree distribution is binomial, or Poisson in the limit of large graph size. Real-world networks are mostly found to be very unlike the random graph in their degree distributions. Far from having a Poisson distribution, the degrees of the vertices in most networks are highly right-skewed, meaning that their distribution has a long right tail of values that are far above the mean.

An alternative way of presenting degree data is to make a plot of the

cumulative distribution function

$$P_k = \sum_{k'=k}^{\infty} p_{k'} \,,$$

which is the probability that the degree is greater than or equal to k. Fig. 1.17 shows cumulative distributions of degree for a number of the networks described in Sec. 1.2: it is easy to see that the distributions are indeed all right-skewed. In particular, of these networks, three of them, (c), (d) and (f), appear to have power-law degree distributions, as indicated by their approximately straight-line forms on the doubly logarithmic scales, and one (b) has a power-law tail but deviates markedly from power-law behavior for small degree. Network (e) has an exponential degree distribution (note the log-linear scales used in this panel) and network (a) appears to have a truncated power-law degree distribution of some type, or possibly two separate power-law regimes with different exponents. Many of them follow power laws in their tails: $p_k \sim k^{-\alpha}$ for some constant exponent α. Note that such power-law distributions show up as power laws in the cumulative distributions also, but with exponent $\alpha - 1$ rather than α:

$$P_k \sim \sum_{k'=k}^{\infty} k'^{-\alpha} \sim k^{-(\alpha-1)} \,.$$

Some of the other distributions have exponential tails: $p_k \sim e^{-k/\kappa}$. These also give exponentials in the cumulative distribution, but with the same exponent:

$$P_k \sim \sum_{k'=k}^{\infty} e^{-k'/\kappa} \sim e^{-k/\kappa} \,.$$

This makes power-law and exponential distributions particularly easy to spot experimentally, by plotting the corresponding cumulative distributions on logarithmic scales (for power laws) or semi-logarithmic scales (for exponentials).

For other types of networks degree distributions can be more complicated. For bipartite graphs (Sec. 1.1), for instance, there are two degree distributions, one for each type of vertex. For directed graphs each vertex has both an

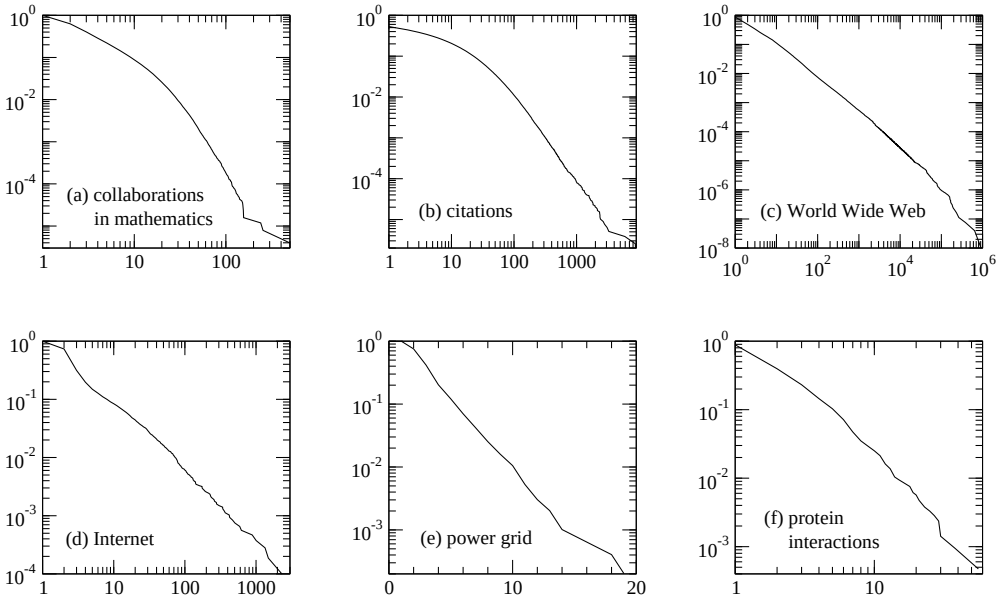

Figure 1.17: Cumulative degree distributions for six different networks. The horizontal axis for each panel is vertex degree k (or in-degree for the citation and Web networks, which are directed) and the vertical axis is the cumulative probability distribution of degrees, i.e., the fraction of vertices that have degree greater than or equal to k. The networks shown are: (a) the collaboration network of mathematicians [53]; (b) citations between 1981 and 1997 to all papers cataloged by the Institute for Scientific Information [103]; (c) a 300 million vertex subset of the World Wide Web, circa 1999 [24]; (d) the Internet at the level of autonomous systems, April 1999 [26]; (e) the power grid of the western United States [126]; (f) the interaction network of proteins in the metabolism of the yeast S. Cerevisiae [61].

in-degree and an out-degree, and the degree distribution therefore becomes a function p_{jk} of two variables, representing the fraction of vertices that simultaneously have in-degree j and out-degree k. In empirical studies of directed graphs like the Web, researchers have usually given only the individual distributions of in- and out-degree [10, 21, 24], i.e., the distributions derived by summing p_{jk} over one or other of its indices. This however discards much of the information present in the joint distribution. It has been found that in- and out-degrees are quite strongly correlated in some networks [94], which suggests that there is more to be gleaned from the joint distribution than is normally appreciated.

1.3.9 Network resilience

Related to degree distributions is the property of resilience of networks to the removal of their vertices, which has been the subject of a good deal of attention in the literature. Most of the considered networks rely for their function on their connectivity, i.e., the existence of paths leading between pairs of vertices. If vertices are removed from a network, the typical length of these paths will increase, and ultimately vertex pairs will become disconnected and communication between them through the network will become impossible. Networks vary in their level of resilience to such vertex removal.

There are also a variety of different ways in which vertices can be removed and different networks show varying degrees of resilience to these also. For example, one could remove vertices at random from a network, or one could target some specific class of vertices, such as those with the highest degrees. Network resilience is of particular importance in epidemiology, where "removal" of vertices in a contact network might correspond for example to vaccination of individuals against a disease. Because vaccination not only prevents the vaccinated individuals from catching the disease but may also destroy paths between other individuals by which the disease might have spread, it can have a wider reaching effect than one might at first think, and careful

consideration of the efficacy of different vaccination strategies could lead to substantial advantages for public health.

Recent interest in network resilience has been sparked by the work of Albert et al. [11], who studied the effect of vertex deletion in two example networks, a 6000-vertex network representing the topology of the Internet at the level of autonomous systems, and a 326 000-page subset of the World Wide Web. Both of the Internet and the Web have been observed to have degree distributions that are approximately power-law in form. The authors measured average vertex-vertex distances as a function of number of vertices removed, both for random removal and for progressive removal of the vertices with the highest degrees.

Fig. 1.18 shows their results for the Internet. They found for both networks

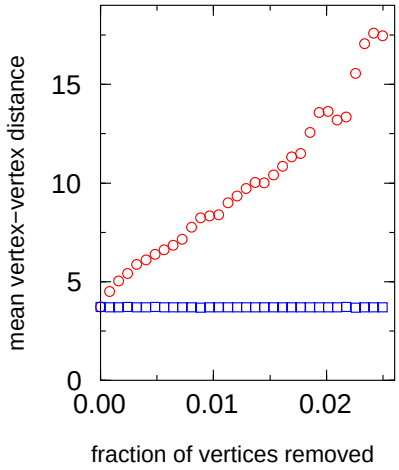

Figure 1.18: Mean vertex-vertex distance on a graph representation of the Internet at the autonomous system level, as vertices are removed one by one. If vertices are removed in random order (squares), distance increases only very slightly, but if they are removed in order of their degrees, starting with the highest degree vertices (circles), then distance increases sharply. After Albert et al. [11].

that distance was almost entirely unaffected by random vertex removal, i.e.,

the networks studied were highly resilient to this type of removal. This is intuitively reasonable, since most of the vertices in these networks have low degree and therefore lie on few paths between others; thus their removal rarely affects communications substantially. On the other hand, when removal is targeted at the highest degree vertices, it is found to have devastating effect. Mean vertex-vertex distance increases very sharply with the fraction of vertices removed, and typically only a few percent of vertices need be removed before essentially all communication through the network is destroyed. Albert et al. expressed their results in terms of failure or sabotage of network nodes. The Internet (and the Web) they suggest, is highly resilient against the random failure of vertices in the network, but highly vulnerable to deliberate attack on its highest degree vertices.

Similar results to those of Albert et al. were found independently by Broder et al. [24] for a much larger subset of the Web graph. Interestingly, however, Broder et al. gave an entirely opposite interpretation of their results. They found that in order to destroy connectivity in the Web one has to remove all vertices with degree greater than five, which seems like a drastic attack on the network, given that some vertices have degrees in the thousands. They thus concluded that the network was very resilient against targeted attack. In fact, however, there is not such a conflict between these results as at first appears. Because of the highly skewed degree distribution of the Web, the fraction of vertices with degree greater than five is only a small fraction of all vertices.

Following these studies, many authors have looked into the question of resilience for other networks. In general the picture seems to be consistent with that seen in the Internet and Web. Most networks are robust against random vertex removal but considerably less robust to targeted removal of the highest degree vertices. Jeong et al. [61] have looked at metabolic networks, Dunne et al. [42, 43] at food webs, Newman et al. [94] at email networks, and a variety of authors at resilience of model networks [11, 25, 29, 30, 55]. A particularly thorough study of the resilience of both real-world and model

networks has been conducted by Holme et al. [55], who looked not only at vertex removal but also at removal of edges, and considered some additional strategies for selecting vertices based on so-called "betweenness" or "load". The edge (node) load is the number of shortest paths (over all pairs of nodes of the network) passing through the edge (node) [34, 92]. In [34] the authors use the global efficiency [72] to evaluate how well a system works before and after the removal of a set of nodes.

1.3.10 Network navigation

Stanley Milgram's famous small-world experiment (Sec. 1.2), in which letters were passed from person to person in an attempt to get them to a desired target individual, showed that there exist short paths through social networks between apparently distant individuals. However, there is another conclusion that can be drawn from this experiment which Milgram apparently failed to notice; it was pointed out in 2000 by Kleinberg [67, 68]. Milgram's results demonstrate that there exist short paths in the network, but they also demonstrate that ordinary people are good at finding them. This is, upon reflection, perhaps an even more surprising result than the existence of the paths in the first place. The participants in Milgram's study had no special knowledge of the network connecting them to the target person. Most people know only who their friends are and perhaps a few of their friends' friends. Nonetheless it proved possible to get a message to a distant target in only a small number of steps. This indicates that there is something quite special about the structure of the network. On a random graph for instance, as Kleinberg pointed out, short paths between vertices exist but no one would be able to find them given only the kind of information that people have in realistic situations. If it were possible to construct artificial networks that were easy to navigate in the same way that social networks appear to be, it has been suggested they could be used to build efficient database structures or better peer-to-peer computer networks [3, 4, 125].

1.4 Random graphs

In a series of seminal papers in the 1950s and 1960s, Paul Erdős and Alfréd Rényi proposed and studied one of the earliest theoretical models of a network, the random graph [45, 46, 47]. This minimal model consists of N nodes or vertices, joined by links or edges which are placed between pairs of vertices chosen uniformly at random (Fig. 1.19). Erdős and Rényi gave a number of

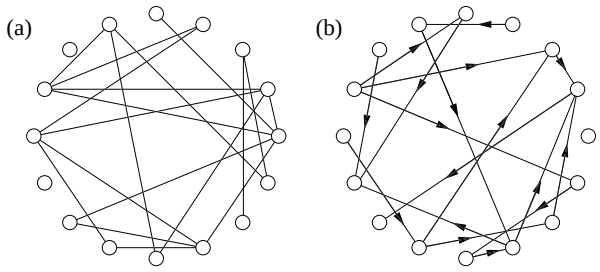

Figure 1.19: (a) A schematic representation of a random graph, the circles representing vertices and the lines edges. (b) A directed random graph, i.e., one in which each edge runs in only one direction [95].

versions of their model. The most commonly studied is the one denoted $G_{N,p}$, in which each possible edge between two vertices is present with independent probability p, and absent with probability $1-p$. Technically, in fact, $G_{N,p}$ is the ensemble of graphs of N vertices in which each graph appears with the probability appropriate to its number of edges: for a graph with N vertices and K edges this probability is $p^K(1-p)^{K_{max}-K}$, where $K_{max} = \frac{1}{2}N(N-1)$.

Often one wishes to express properties of $G_{N,p}$ not in terms of p but in terms of the average degree z of a vertex. The average number of edges on the graph as a whole is $\frac{1}{2}N(N-1)p$, and the average number of ends of edges is twice this, since each edge has two ends. So the average degree of a vertex is

$$z = \frac{N(N-1)p}{N} = (N-1)p \simeq Np\,,$$

where the last approximate equality is good for large N. Thus, once N is

known, any property that can be expressed in terms of p can also be expressed in terms of z.

The Erdős–Rényi (ER) random graph has a number of desirable properties as a model of a network. In particular it is found that many of its ensemble average properties can be calculated exactly in the limit of large N [23, 60]. For example, one interesting feature, which was demonstrated in the original papers by Erdős and Rényi, is that the model shows a phase transition with increasing z at which a giant component forms. A component is a subset of vertices in the graph each of which is reachable from the others by some path through the network. For small values of z, when there are few edges in the graph, it is not surprising to find that most vertices are disconnected from one another, and components are small, having an average size that remains constant as the graph becomes large. However, there is a critical value of z above which the one largest component in the graph contains a finite fraction S of the total number of vertices, i.e., its size NS scales linearly with the size of the whole graph. This largest component is the giant component. In general there will be other components in addition to the giant component, but these are still small, having an average size that remains constant as the graph grows larger. The phase transition at which the giant component forms occurs precisely at $z = 1$.

The formation of a giant component in the random graph is reminiscent of the behavior of many real-world networks. One can imagine loose-knit networks for which there are so few edges that, presumably, the network has no giant component, and all vertices are connected to only a few others. The social network in which pairs of people are connected if they have had a conversation within the last 60 seconds, for example, is probably so sparse that it has no giant component. The network in which people are connected if they have ever had a conversation, on the other hand, is very densely connected and certainly has a giant component.

Random graphs have been employed extensively as models of real-world

networks of various types, particularly in epidemiology. The passage of a disease through a community depends strongly on the pattern of contacts between those infected with the disease and those susceptible to it. This pattern can be depicted as a network, with individuals represented by vertices and contacts capable of transmitting the disease by edges. The large class of epidemiological models known as susceptible/infectious/recovered (or SIR) models [15, 71, 104] makes frequent use of the so-called fully mixed approximation, which is the assumption that contacts are random and uncorrelated, i.e., that they form a random graph.

However, the random graph differs from real-world networks in some fundamental ways also. Two differences in particular have been noted in the recent literature [9, 115]. First, as pointed out by Watts and Strogatz [124, 126] real-world networks show strong clustering or network transitivity, where ER model does not. A network is said to show clustering if the probability of two vertices being connected by an edge is higher when the vertices in question have a common neighbor. That is, there is another vertex in the network to which they are both attached. As described in Sec. 1.3.2, Watts and Strogatz measured this clustering by defining a clustering coefficient C, which is the average probability that two neighbors of a given vertex are also neighbors of one another.

In many real-world networks the clustering coefficient is found to have a high value, anywhere from a few percent to 50 percent or even more. In the ER random graph, on the other hand, the probabilities of vertex pairs being connected by edges are by definition independent, so that there is no greater probability of two vertices being connected if they have a mutual neighbor than if they do not. This means that the clustering coefficient for a random graph is simply $C = p$, or equivalently $C \simeq z/N$. In Table 1.6, clustering coefficients for a number of real-world networks are compared with their values on a random graph with the same number of vertices and edges. The graphs listed in the table are:

Table 1.6: Number of vertices N, mean degree z and clustering coefficient C for a number of different networks [91].

			clustering coefficient C	
network	N	z	measured	random graph
Internet (autonomous systems) [98]	6 374	3.8	0.24	0.00060
World Wide Web (sites) [2]	153 127	35.2	0.11	0.00023
power grid (sites) [126]	4 941	2.7	0.080	0.00054
biology collaborations [87]	1 520 251	15.5	0.081	0.000010
mathematics collaborations [90]	253 339	3.9	0.15	0.000015
film actor collaborations [95]	449 913	113.4	0.20	0.00025
company directors [95]	7 673	14.4	0.59	0.0019
word co-occurrence [58]	460 902	70.1	0.44	0.00015
neural network [126]	282	14.0	0.28	0.049
metabolic network [49]	315	28.3	0.59	0.090
food web [84]	134	8.7	0.22	0.065

- Internet: a graph of the fibre optic connections that comprise the Internet, at the level of so-called "autonomous systems". Examples of autonomous systems might be the computers at a company, a university, or an Internet service provider.

- World Wide Web: a graph of sites on the World Wide Web in which edges represent "hyperlinks" connecting one site to another. A site in this case means a collection of pages residing on a server with a given name. Although hyperlinks are directional, their direction has been ignored in this calculation of the clustering coefficient.

- Power grid: a graph of the Western States electricity transmission grid in the United States. Vertices represent stations and substations; edges represent transmission lines.

- Biology collaborations: a graph of collaborations between researchers working in biology and medicine. A collaboration between two scientists is defined in this case as coauthorship of a paper that was catalogued in the Medline bibliographic database between 1995 and 1999 inclusive.

- Mathematics collaborations: a similar collaboration graph for mathematicians, derived from the archives of *Mathematical Reviews*.

- Film actor collaborations: a graph of collaborations between film actors, where a collaboration means that the two actors in question have appeared in a film together. The data are from the Internet Movie Database.

- Company directors: a collaboration graph of the directors of companies in the Fortune 1000 for 1999. (The Fortune 1000 is the 1000 US companies with the highest revenues during the year in question.) Collaboration in this case means that two directors served on the board of a Fortune 1000 company together.

- Word co-occurrences: a graph in which the vertices represent words in the English language, and an edge signifies that the vertices it connects frequently occur in adjacent positions in sentences.
- Neural network: a graph of the neural network of the worm C. Elegans.
- Metabolic network: a graph of interactions forming a part of the energy generation and small building block synthesis metabolism of the bacterium E. Coli. Vertices represent substrates and products, and edges represent interactions.
- Food web: the food web of predator-prey interactions between species in Ythan Estuary, a marine estuary near Aberdeen, Scotland. Like the links in the World Wide Web graph, the directed nature of the interactions in this food web have been neglected for the purposes of calculating the clustering coefficient.

As the table shows, the agreement between the clustering coefficients in the real networks and in the corresponding random graphs is not good. The real and theoretical figures differ by as much as four orders of magnitude in some cases. Clearly, the random graph does a poor job of capturing this particular property of networks.

A second way in which random graphs differ from their real-world counterparts is in their degree distributions, a point which has been emphasized particularly in [10, 20]. The probability p_k that a vertex in an ER random graph has degree exactly k is given by the binomial distribution:

$$p_k = \binom{N-1}{k} p^k (1-p)^{N-1-k} .$$

In the limit where $N \gg kz$, this becomes

$$p_k = \frac{z^k e^{-z}}{k!} ,$$

which is the well-known Poisson distribution. Both binomial and Poisson distributions are strongly peaked about the mean z, and have a large-k tail that

decays rapidly as $1/k!$. These predictions can be compared with the degree distributions of real networks by constructing histograms of the degrees of vertices in the real networks. Some examples, taken from the networks described above, are shown in Fig. 1.20. As the figure shows, in most cases the degree distribution of the real network is very different from the Poisson distribution. Many of the networks, including Internet and World Wide Web graphs, appear to have power-law degree distributions [10, 24, 48], which means that a small but non-negligible fraction of the vertices in these networks have very large degree. This behavior is quite unlike the rapidly decaying Poisson degree distribution and can have profound effects on the behavior of the network. Other networks, particularly the collaboration graphs, appear to have power-law degree distributions with an exponential cutoff at high degree [14, 87, 89], while others still, such as the graph of company directors, seem to have degree distributions with a purely exponential tail [95]. The power grid of Table 1.6 is another example of a network that has an exponential degree distribution [14].

The random graph reproduces well one of the principal features of real-world networks discussed in Sec. 1.3, namely the small-world effect. The mean number of neighbors a distance d away from a vertex in a random graph is z^d, and hence the value of d needed to encompass the entire network is $z^L \simeq N$. Thus a typical distance through the network is $L = \log N / \log z$, which satisfies the definition of the small-world effect given in Sec. 1.3.1.

It is possible to generalize the ER random graph to mimic the clustering and degree properties of real-world networks. In the last few years, an elegant body of theory for the correction of the degree distribution has been developed. There exist also methods that allow to introduce clustering into random graphs; however, work on this latter problem is significantly less far advanced than work on degree distributions (at present, only a few preliminary results have been obtained). Whether these results can be extended, and how, are open questions.

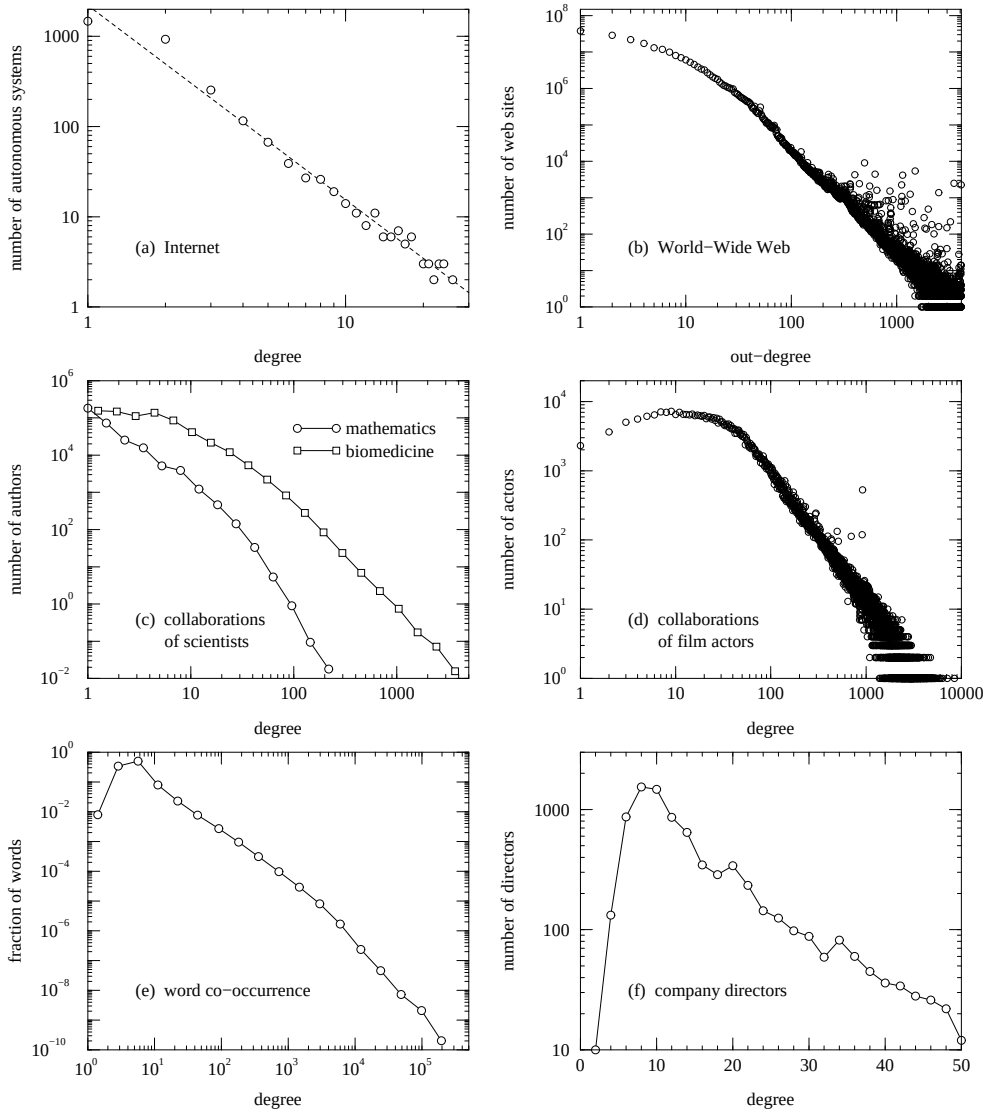

Figure 1.20: Measured degree distributions for a number of different networks. (a) Physical connections between autonomous systems on the Internet, circa 1997 [48]. (b) A 200 million page subset of the World Wide Web, circa 1999 [24]. The figure shows the out-degree of pages, i.e., numbers of links pointing from those pages to other pages. (c) Collaborations between biomedical scientists and between mathematicians [87, 90]. (d) Collaborations of film actors [14]. (e) Co-occurrence of words in the English language [58]. (f) Board membership of directors of Fortune 1000 companies for year 1999 [95].

1.5 Scale-free networks

Networks with power-law degree distributions have been the focus of a great deal of attention in the literature [9, 40, 115]. They are sometimes referred to as scale-free networks [20], although it is only their degree distributions that are scale-free; one can and usually does have scales present in other network properties. The earliest published example of a scale-free network is probably Price's network of citations between scientific papers [37]. He quoted a value of $\alpha = 2.5$ to 3 for the exponent of his network. In a later paper he quoted a more accurate figure of $\alpha = 3.04$ [38]. He also found a power-law distribution for the out-degree of the network (number of bibliography entries in each paper), although later work has called this into question [120]. More recently, power-law degree distributions have been observed in a host of other networks, including notably other citation networks [103, 108], the World Wide Web [10, 21, 24], the Internet [26, 48, 121], metabolic networks [61, 62], telephone call graphs [5, 6] and the network of human sexual contacts [63, 78]. The degree distributions of some of these networks are shown in Fig. 1.17.

Other common functional forms for the degree distribution are exponentials, such as those seen in the power grid [14] and railway networks [109], and power laws with exponential cutoffs, such as those seen in the network of movie actors [14] and some collaboration networks [89].

In [20] the authors show that a common property of many large networks is that the vertex degrees follow a scale-free power-law distribution. This feature was found to be a consequence of two generic mechanisms:

1. Networks expand continuously by the addition of new vertices;

2. New vertices attach preferentially to sites that are already well connected (the "rich-get-richer" phenomenon).

A model based on these two ingredients reproduces the observed stationary scale-free distributions, which indicates that the development of large networks

is governed by robust self-organizing phenomena that go beyond the particulars of the individual systems. Exploring several large databases describing the topology of large networks that span fields as diverse as the WWW or citation patterns in science, the authors show that, independent of the system and the identity of its constituents, the probability p_k that a vertex in the network interacts with k other vertices decays as a power law, following $p_k \sim k^{-\gamma}$. This result indicates that large networks self-organize into a scale-free state, a feature unpredicted by all existing random network models. To explain the origin of this scale invariance, they show that existing network models fail to incorporate growth and preferential attachment, two key features of real networks. Using a model incorporating these two ingredients, the authors show that they are responsible for the power-law scaling observed in real networks. Moreover, the authors also argue that these ingredients play an easily identifiable and important role in the formation of many complex systems.

A common feature of the ER and WS models is that the probability of finding a high degree vertex (that is, with a large k) decreases exponentially with k; thus, vertices with high degree are practically absent. In contrast, the power-law tail characterizing p_k for many real-world networks indicates that high degree (large k) vertices have a large chance of occurring. There are two generic aspects of real networks that are not incorporated in these models. First, both models assume that one starts with a fixed number N of vertices that are then randomly connected (ER model), or reconnected (WS model), without modifying N. In contrast, most real-world networks are open and they form by the continuous addition of new vertices to the system, thus the number of vertices N increases throughout the lifetime of the network. For example, the actor network grows by the addition of new actors to the system, the WWW grows exponentially over time by the addition of new Web pages [75, 76] and the research literature constantly grows by the publication of new papers. Consequently, a common feature of these systems is that the network

continuously expands by the addition of new vertices that are connected to the vertices already present in the system.

Second, the random network models assume that the probability that two vertices are connected is random and uniform. In contrast, most real networks exhibit preferential connectivity. For example, a new actor is most likely to be cast in a supporting role with more established and better-known actors. Consequently, the probability that a new actor will be cast with an established one is much higher than that the new actor will be cast with other less-known actors. Similarly, a newly created Web page will be more likely to include links to well-known popular documents with already-high connectivity, and a new manuscript is more likely to cite a well-known and thus much-cited paper than its less-cited and consequently less-known peer. These examples indicate that the probability with which a new vertex connects to the existing vertices is not uniform; there is a higher probability that it will be linked to a vertex that already has a large number of connections.

Some networks evolve not only by adding new vertices but by adding (and sometimes removing) connections between established vertices. Although these and other system-specific features could modify the exponent γ, the model proposed in [20] offers the first successful mechanism accounting for the scale-invariant nature of real networks. Growth and preferential attachment are mechanisms common to a number of complex systems, including business networks [18], social networks (describing individuals or organizations), transportation networks [19], etc. Scale-invariant state could be a generic property of many complex networks. Possible scale-free features of genetic and signaling networks, for example, could reflect the networks' evolutionary history, dominated by growth and aggregation of different constituents, leading from simple molecules to complex organisms. Similar mechanisms could explain the origin of the social and economic disparities governing competitive systems, because the scale-free inhomogeneities are the inevitable consequence of self-organization due to the local decisions made by the individual vertices,

based on information that is biased towards the more visible (richer) vertices, irrespective of the nature and origin of this visibility.

1.6 A simulation tool for the analysis of complex networks

A weighted random digraph is used as network model. The graph contains N nodes; each edge is present with probability p and has weight w_1, w_2 or w_3 with probability p_{w_1}, p_{w_2} or p_{w_3} respectively. The graph is represented by the adjacency matrix $[a_{ij}]$ and the matrix of weights $[w_{ij}]$; it is also assumed that the graph cannot contain parallel edges nor self-loops. As for the cost evaluator function, assuming that w_{ij} represents the time a packet of information needs to be transmitted from node i to node j along the link connecting them, $\gamma(w_{ij}) = 1/w_{ij}$: the fastest links are also the most expensive ones. The matrix of the shortest paths d_{ij} is computed by using Dijkstra's algorithm.

A source node is a vertex that has only outgoing edges, a sink node is a vertex that has only incoming edges and an isolated node is a vertex that has no edges at all. A directed graph is *strongly connected* if for each couple of vertices i and j there exist at least a path from i to j and at least a path from j to i. A necessary (but not sufficient) condition for the strong connection property to hold is that the graph has no source, sink or isolated vertices. If this property does not hold, the strongly connected components are determined. A question then arises as to which vertices play a fundamental role in graph connectivity. The relevance of a node is evaluated using two different measures:

- The in- and out-degree of a node, i.e., the number of incoming and outgoing edges respectively;
- The load of a node, i.e., the number of shortest paths – between all pairs of nodes of the network – that pass through the node.

According to the nomenclature adopted in [34], failure is the removal of

randomly selected nodes and attack is the targeted removal of the most important nodes (detected using one of the two previous quantities). In this work, it is assumed that the network is only subject to failures and that the probability of having two or more damaged nodes at the same time is negligible. When a failure occurs, the efficiency of the resulting graph (containing $N-1$ nodes) is computed, so that it can be compared with the efficiency of the graph before the failure occurred. It is clear that removing the vertex with the highest load (and all its incoming and outgoing edges) could cause a drastic efficiency reduction, eventually leading to the formation of two or more components, making the communication between some pairs of nodes impossible (Fig. 1.21). Similarly, one may be interested in determining the most

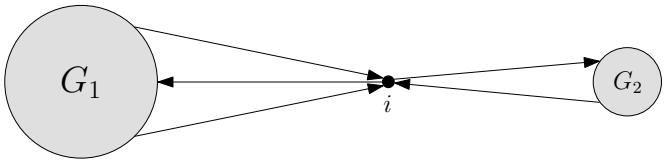

Figure 1.21: The removal of node i prevents the communication between the nodes in the subgraph G_1 and the nodes in the subgraph G_2.

important edges: removing the edge with the highest load may also cause a large efficiency reduction, eventually leading to the formation of two or more components, again making the communication between some pairs of nodes difficult or even impossible.

Figs. 1.22–1.28 show the simulation results for a graph containing $N = 100$ nodes and characterized by the following probabilities: $p = 0.1$; $p_{w_1} = 0.6$, $p_{w_2} = 0.3$, $p_{w_3} = 0.1$, where $w_1 = 1$, $w_2 = 2$ and $w_3 = 3$. Moreover, a node failure occurs with probability 0.1. The statistical properties have been computed considering $q = 100$ realizations of the graph.

Fig. 1.22 shows the empirical values of p obtained in each realization of the graph and its expected value. Fig. 1.23 is a synoptic picture providing several information on graph topology. The legend for the y axis is:

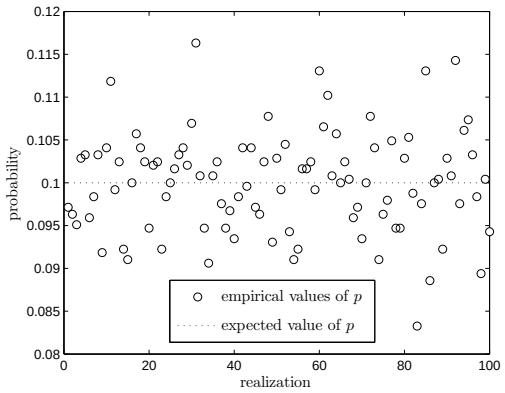

Figure 1.22: Empirical values of p obtained in each realization of the graph and its expected value.

- SCP: strong connection property (• holds, + does not hold)
- SoN: source nodes (• none, + at least one)
- SiN: sink nodes (• none, + at least one)
- IsN: isolated nodes (• none, + at least one)
- NF: node failures (• none, + a node failure occurred)

Fig. 1.23 allows to know if in a particular realization there were source nodes and/or sink nodes and/or isolated nodes. In this case, for example, there were 20 realizations containing at least a source node, 24 realizations containing at least a sink node and no realizations containing isolated nodes. It also allows to find out if the graph was or not strongly connected and if a node failure occurred; in particular, there were 38 realizations in which the graph was not strongly connected and 13 node failures occurred. Fig. 1.24 completes Fig. 1.23, providing further information such as the number of source, sink and isolated nodes in each realization, respectively from top to bottom. In this case, there were no isolated nodes and never more than two source/sink nodes in each realization.

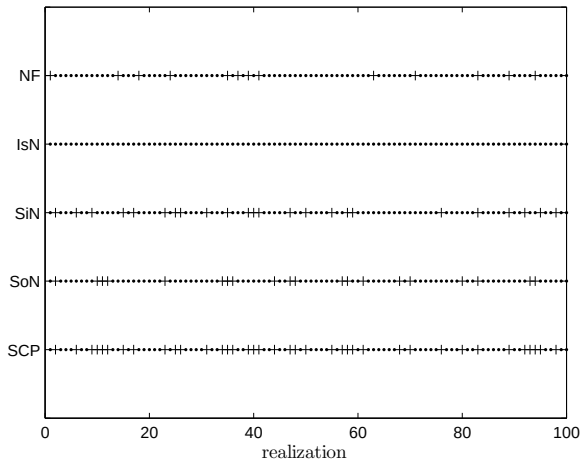

Figure 1.23: A synoptic picture of graph topology.

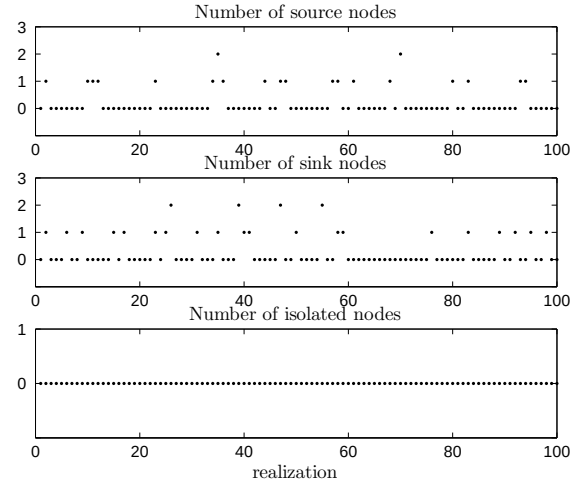

Figure 1.24: Number of source, sink and isolated nodes in each realization, respectively from top to bottom.

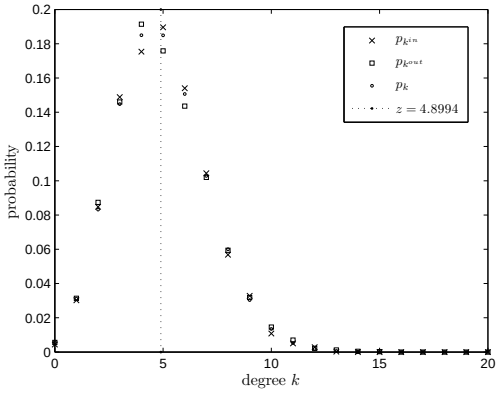

Figure 1.25: In-degree, out-degree and binomial probability distributions ($p_{k^{in}}$, $p_{k^{out}}$ and p_k respectively).

Fig. 1.25 shows the in-degree probability distribution ($p_{k^{in}}$), the out-degree probability distribution ($p_{k^{out}}$) and the binomial distribution (p_k): a generic value $p_{\bar{k}^{in}}$ ($p_{\bar{k}^{out}}$) extracted from the in- (out-) degree distribution represents the probability that a randomly chosen vertex has in- (out-) degree $\bar{k}^{in}$ ($\bar{k}^{out}$). The resemblance between the three distributions appears evident from the figure, as expected. The mean degree z (vertical dotted line) is also indicated, along with its value.

Fig. 1.26a shows the joint probability distribution p_{jk}, i.e., the probability to find a node with in-degree j and out-degree k. Fig. 1.26b contains the joint probability binomial distribution, obtained assuming a binomial distribution for both the in- and out- degree distributions. The resemblance with the empirical distribution shown in Fig. 1.26a is quite evident.

Fig. 1.27a shows the node load distribution d_{nl}, i.e., the probability to find a node with a specified number of shortest paths passing through it. This distribution is characterized by two "peaks": the first one ($\simeq 0.0162$) is the maximum value of the distribution and is centered in 0, the second one ($\simeq 0.0106$) corresponds to a node load equal to 40. There are no vertices having more than 482 shortest paths passing through them. Fig. 1.27b shows

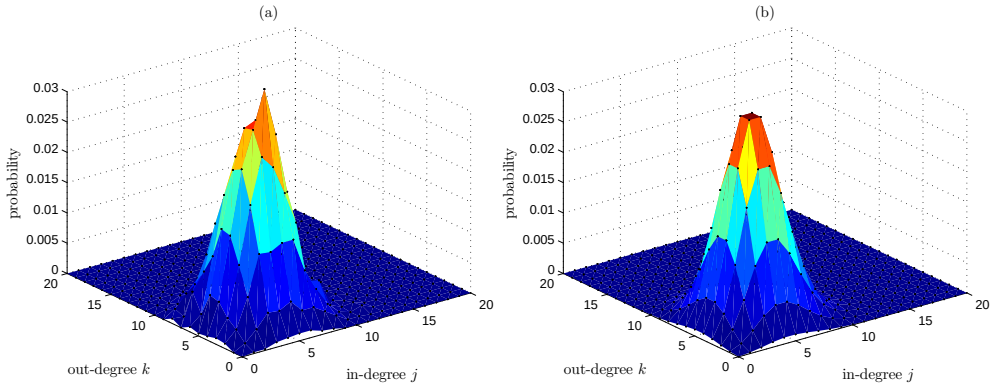

Figure 1.26: (a) Joint probability distribution p_{jk}. (b) Joint probability binomial distribution.

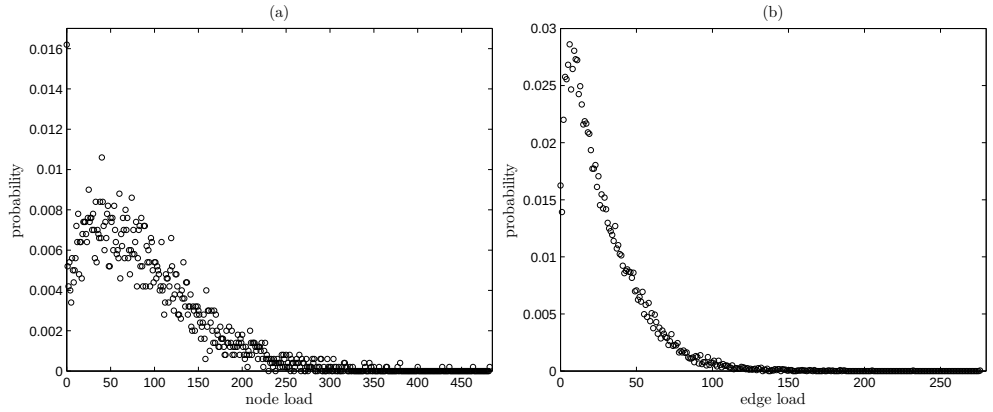

Figure 1.27: (a) Node load distribution d_{nl}. (b) Edge load distribution d_{el}.

the edge load distribution d_{el}, i.e., the probability to find an edge with a certain number of shortest paths passing through it. Similarly to the node load distribution, this distribution also presents two "peaks": the first one ($\simeq 0.0162$) is centered in 0, the second one ($\simeq 0.0286$) corresponds to an edge load equal to 6 and is also the maximum value of the distribution. There are no edges having more than 276 shortest paths passing through them.

Fig. 1.28 shows the shortest path distribution d_{sp}, i.e., the probability with

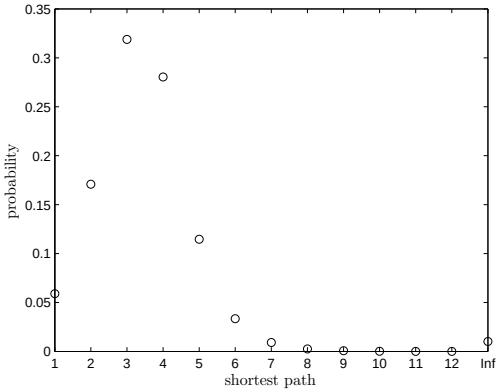

Figure 1.28: Shortest path distribution d_{sp}.

which two randomly chosen nodes are connected by a shortest path of a given length. It is very likely to find shortest paths whose length is 3 or 4, while it is very unlikely to find shortest paths of length greater than 6. Moreover, given two randomly chosen nodes i and j, the probability with which there exists no path from i to j is $1 - 0.9898 = 0.0102$.

Table 1.7 contains some statistical properties of the considered graph. The diameter of the graph is the maximum number of edges in the shortest paths. It is worth noting that, even if the probability of the graph being strongly connected is 0.62 (since there were 38 realizations in which the graph was not strongly connected), the probability that there is at least a path between two randomly chosen nodes is 0.9898 and the probability that there is also at least an inverse path is 0.9797.

Table 1.7: E_{glob}, E_{loc}, $Cost$, diameter and other statistical properties of the considered graph.

Number of node failures occurred	13
Global efficiency of the resulting graph after a node failure	0.4264
Number of times the graph lost the strong connection property after a node failure	1
Number of times the resulting graph was strongly connected after a node failure	6
Probability with which two randomly chosen vertices are connected by at least a path	0.9898
Probability with which two randomly chosen vertices are connected by at least a path in both directions	0.9797
Diameter	6.1600
E_{glob}	0.4414
E_{loc}	0.1079
$Cost$	0.0999

1.7 Conclusions

The last few years have witnessed a tremendous activity devoted to the characterization and understanding of networked systems. Indeed large complex networks arise in a vast number of natural and artificial systems. Ecosystems consist of species whose interdependency can be mapped into intricate food webs. Social systems may be represented by graphs describing various interactions among individuals. The Internet and the World Wide Web are prototypical examples of self-organized networks emerging in the technological world. Large infrastructures such as power grids and the air transportation network are critical networked systems of our modern society. Finally, the living cell is not an exception either, its organization and function being the outcome of a complex web of interactions among genes, proteins and other molecules.

For a long time all these systems have been considered as haphazard set of points and connections, mathematically framed in the random graph paradigm. This situation has radically changed in the last five years, during which the study of complex networks has received a boost from the ever-increasing availability of large data sets and the increasing computer power for storage and manipulation. In particular, mapping projects of the World Wide Web and the physical Internet offered the first chance to study the topology of large complex networks. Gradually, other maps followed describing many networks of practical interest in social science, critical infrastructures and biology. Researchers thus have started to have a systematic look at these large data sets, searching for hidden regularities and patterns that can be considered as manifestations of underlying laws governing the dynamics and evolution of these complex systems. Indeed, when studying the structure of complex networks, one finds out that in spite of the apparent complexity and randomness of the underlying systems, clear patterns and regularities emerge, which can be expressed in mathematical and statistical fashion.

Specifically, many of these systems show the small-world property, which

implies that in the network the average topological distance between the various nodes increases very slowly with the number of nodes (logarithmically or even slower). A particularly important finding is the realization that many networks are characterized by the statistical abundance of "hubs", i.e., nodes with a large number of connections to other elements. This feature has its mathematical roots in the observation that the number of elements with k links follows a power-law distribution, indicating the lack of any characteristic scale. This has allowed the identification of the class of scale-free networks whose topological features turn out to be extremely relevant in assessing the physical properties of the system as a whole, such as its robustness to damages or vulnerability to malicious attack.

The attempt to model and understand the origin of the observed topological properties of real networks has led to a radical change of perspective, shifting the focus from static graphs, aiming to reproduce the structure of the network in a certain moment, to modeling network evolution. This new approach is the outcome of the realization that most complex networks are the result of a growth process. As a result, networks are currently viewed as dynamical systems that evolve through the subsequent addition and deletion of vertices and edges. The set of dynamical rules defining these processes thus outlines the dynamical theory required in order to understand the macroscopic properties of networks. This methodology that is akin to the statistical physics approach to complex phenomena appears as a revolutionary path in our understanding of networked systems and provides new techniques to approach conceptual and practical problems in this field.

While in the past few years the advances in network research were truly amazing, both in their impact on basic science and practical implications, they have highlighted the incompleteness of our knowledge as well [12].

Chapter 2

Identification of complex networks by the method of stages

In this chapter, a weighted directed random graph is used as network model. The graph contains a fixed number N of nodes and a variable number of edges: in particular, each edge is present with probability p. Some statistical properties (such as strong connection, global and local efficiency, cost, etc) are computed and their reliance on probability p is studied. Some probability distributions (such as shortest path, edge/node load) are also drawn and, by using the method of stages, the best fitting curves are computed. Finally, the way as parameters characterizing such curves change when p varies is investigated. The general structure of the proposed fitting technique allows to model several aspects of complex networks and makes possible its use in many different fields. For the reader's convenience, the introduction provides a short overview on complex networks; moreover, the concepts of efficiency and cost of a network, illustrated in Sec. 1.3.3 and Sec. 1.3.5 respectively, are briefly recalled. The content of this chapter is based on [16].

2.1 Introduction

A network can be mathematically represented by a graph [116] in which vertices represent systems and edges represent the relationships between them. A complex network [9, 13, 41, 74, 92] is a network involving a big number of systems interacting each other.

Examples of real-world networks are the Internet, the World Wide Web, the electrical power grid, social networks of acquaintance or other connections between individuals, neural networks, metabolic networks, food webs, etc. Most of these networks exhibit the so called *small-world* effect [126]: most pairs of vertices are connected by a short path through the network. The small-world effect was empirically observed in social systems by Stanley Milgram in the 1960s [82, 119]. One of his famous experiments showed that there exist short paths through social networks between apparently distant individuals. Precisely, in this experiment letters passed from person to person were able to reach a designated target individual in only six steps: thereof the popular concept of the "six degrees of separation" [92]. It is clear that the small-world effect strongly affects the dynamics of processes taking place on networks, such as the flow of traffic on the Internet or the spread of a disease over a social network [92].

A random graph [23, 45, 91, 113] consists of N nodes or vertices joined by links or edges, each of which is present with probability p. It is characterized by a binomial degree distribution and shows the small-world effect. Random graphs have been employed extensively as models of real-world networks of various types, particularly in epidemiology [71, 95, 104]. A number of real-world networks, as for example the World Wide Web [10], collaborations of scientists [89] and movie actors [126], have power-law degree distributions, i.e., the degrees of the vertices are highly right-skewed. These networks are sometimes referred to as *scale-free* networks [9, 20, 92]. Random graphs do not show strong *clustering* or *network transitivity* (a network is said to show clustering if the probability of two vertices being connected by an edge is higher

when the vertices in question have a common neighbor [91]). The probabilities of vertex pairs being connected by edges are by definition independent, so that there is no greater probability of two vertices being connected if they have a mutual neighbor than if they do not [91]. Random graphs can be generalized to mimic the clustering and degree properties of real-world networks [83, 90, 91, 95].

In the present chapter, a generic network is represented by means of a weighted directed random graph containing a fixed number N of nodes and a variable number of edges, each of which is present with probability p. Some statistical properties (such as strong connection, global and local efficiency, cost, etc) are computed and their reliance on probability p is studied. Some probability distributions (such as shortest path, edge/node load) are also drawn and, by using the method of stages, the best fitting curves are computed. Finally, the way as parameters characterizing such curves change when p varies is investigated.

For example, referring to a communication network, nodes can represent systems and edges can represent communication channels through which systems exchange information. If this exchange is not always allowed to be bidirectional, the network has to be represented by a directed graph (or digraph, for short). Each channel is characterized by a particular delay whose value is the weight of the corresponding edge in the graph. Therefore, a communication network can be effectively represented by a weighted random digraph.

The outline of this chapter is as follows. In Sec. 2.2, after introducing the mathematical tools used for graph representation, the concepts of strong connection, efficiency and cost of a network are illustrated. The meanings of the shortest path distribution and edge/node load distributions are also discussed. Sec. 2.3 describes the algorithm, based on the method of stages, used to find the best fitting curves for the distributions previously introduced. Sec. 2.4 presents a case study showing the experimental results of computer simulations. Finally, Sec. 2.5 contains some concluding remarks.

2.2 Description of the network properties that will be computed in simulations

A weighted random digraph can be described by the *adjacency matrix* $A = [a_{ij}]$, whose generic element a_{ij} is 1 if there exists a directed edge connecting i to j and 0 otherwise, and the *matrix of weights* $W = [w_{ij}]$, whose generic element w_{ij} is the weight associated with the edge connecting i to j. The weight of an edge in a directed graph is often thought of as its length and, accordingly, the length of a path is just defined as the sum of the weights on the relevant edges. It is assumed that the graph cannot contain parallel edges nor self-loops. It is important to notice that if there exists a directed edge connecting node i to node j, it does not necessarily coincide with the shortest path between them: in other words, in general w_{ij} do not satisfy the triangle inequality ($w_{ij} \leq w_{ih} + w_{hj} \; \forall \, i, h, j$). By using the information contained in matrices A and W and applying Dijkstra's algorithm [72, 88, 116], the matrix $D = [d_{ij}]$ containing the lengths of the shortest paths between all pairs of vertices is computed.

Referring to a communication network, if the efficiency of the channel connecting node i to node j is defined to be inversely proportional to the time an information packet needs for going from i to j, running along the shortest path between them, that is, $\epsilon_{ij} = 1/d_{ij}$, the average *efficiency* of the graph (1.4) can be defined as [72]

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

The *global efficiency* of the graph, E_{glob}, is obtained by dividing E by the efficiency of the complete graph, that is, the graph containing all the $N(N-1)$ possible edges [72]: obviously, E_{glob} lies in the range $[0, 1]$.

It is also possible to compute the efficiency of the subgraph containing the nodes having an incoming edge from a particular node of the graph and all the edges between them. In order to get a value that lies in the range $[0, 1]$, the

efficiency is divided by the efficiency of the corresponding complete subgraph. The *local efficiency* [72] is the average of the N values obtained in this way. If a node has out-degree less than two, the efficiency of its subgraph is set to zero.

The *cost* of a network (1.5) is defined as [74]

$$Cost = \frac{\sum_{i \neq j} a_{ij}\, \gamma(w_{ij})}{\sum_{i \neq j} \gamma(w_{ij})}\,,$$

where $\gamma(w_{ij})$ is the cost associated with the edge whose initial and terminal vertices are respectively i and j. With reference again to a communication network, it is reasonable to set $\gamma(w_{ij}) = 1/w_{ij}$: the fastest links are also the most expensive ones. Since the denominator of (1.5) is the cost of the complete graph, $Cost$ lies in the range $[0, 1]$.

Another important graph property is the *strong connection*, that is, the existence of directed paths between all pairs of vertices. A necessary (but not sufficient) condition for the strong connection property to hold is the absence of source, sink or isolated vertices. If this property does not hold, the strongly connected components [116] are determined. By using the adjacency matrix, it is easy to establish if the graph is strongly connected and also compute the diameter of the corresponding unweighted digraph, that is, the number of edges in the longest shortest path. The procedure is as follows. The matrix $\bar{A}_1 = A + I$ is computed, where I is the identity matrix: if all the elements of $\bar{A}_1$ are equal to one, then the graph is strongly connected and its diameter is one (i.e., the graph is complete). Otherwise, if $\bar{A}_1$ contains some zeros, $\bar{A}_1^2$ is computed and all the elements greater than zero are set to one, in order to get a logical matrix $\bar{A}_2$ containing only zeros and ones. If all the elements of $\bar{A}_2$ are equal to one, then the graph is strongly connected and its diameter is two; if this is not the case, there are two possibilities: if $\bar{A}_2 = \bar{A}_1$, then the graph is not strongly connected and its diameter is one, otherwise process goes on computing $\bar{A}_1^3$ and the corresponding logical matrix $\bar{A}_3$, and so on. In order to compute the diameter of the weighted digraph, firstly the shortest paths

between all pairs of vertices (using, for example, Dijkstra's algorithm) have to be found and then the number of edges in each path has to be calculated: the maximum of the values obtained in this way is the diameter of the graph.

The computation of some probability distributions allows to get a deeper insight of the network. Three interesting probability distributions are based on the concepts of shortest path and edge/node load (where edge (node) load is the number of shortest paths passing through the edge (node) [34, 92]):

- The shortest path distribution d_{sp}, that is, the probability with which two randomly chosen nodes are separated by a shortest path of a given length;
- The edge load distribution d_{el}, that is, the probability of having a certain number of shortest paths passing through a generic edge;
- The node load distribution d_{nl}, that is, the probability of having a certain number of shortest paths passing through a generic node.

These probability distributions could be usefully utilized, for example, in congestion control problems or in more general problems concerning networked control systems.

2.3 Distribution fitting with the method of stages

In the present section, the problem of finding the best fitting curve for a given probability distribution is studied. The set $\mathcal{Y} = \{y_0, \ldots, y_T\}$ contains the values of the probability distribution, while $\mathcal{T} = \{0, \ldots, T\}$ contains the values of the independent discrete random variable (events associated with values greater than T have probability 0 to happen). The goal is to find a probability density function of the form

$$f_r(t; k_1, \ldots, k_r; a_1, \ldots, a_r) = \sum_{h=1}^{r} k_h a_h^h \frac{t^{h-1}}{(h-1)!} e^{-a_h t}\,, \tag{2.1}$$

where $t \in \mathbb{R}$, $t \geq 0$ is the continuous random variable and $k_h, a_h \in \mathbb{R}$, $0 \leq k_h \leq 1$, $a_h \geq 0$, $(h = 1, \ldots, r)$ are parameters such that the least square error

$$e_r(k_1, \ldots, k_r; a_1, \ldots, a_r) = \sum_{i=0}^{T} (f_r(i) - y_i)^2$$

is minimized and

$$\sum_{h=1}^{r} k_h = 1 \,. \tag{2.2}$$

Equivalently, for each value of r, the problem is to find the optimal parameters k_h^o and a_h^o $(h = 1, \ldots, r)$ characterizing the best fitting curve f_r^o and satisfying constraint (2.2).

Eq. (2.1) represents the parallel-series connection shown in Fig. 2.1. The

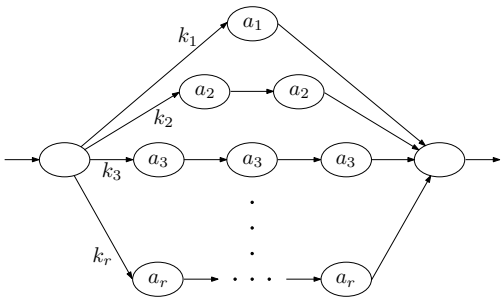

Figure 2.1: Stage diagram relative to Eq. (2.1).

$r(r+1)/2$ stages are statistically independent of each other and the probability density function of the stage a_h is $a_h\, e^{-a_h t}$ $(h = 1, \ldots, r)$. The series connection represents the sum of random variables, while the parallel connection represents a choice: constraint (2.2) directly follows by one of the probability axioms. The characteristic function related to the stage diagram of Fig. 2.1 (i.e., the Laplace transform of (2.1)) is [118]

$$F_r(s) = \mathcal{L}[f_r(t)] = \sum_{h=1}^{r} k_h a_h^h \frac{1}{(s + a_h)^h} \,,$$

from which it is clear that constraint (2.2) is practically automatically satisfied

if (2.1) is a very good interpolation of the probability distribution, since

$$F_r(0) = \int_0^\infty f_r(t)\,\mathrm{d}t = 1\,. \tag{2.3}$$

In other words, the more (2.1) well fits the distribution, the better (2.2) is satisfied (the value of the integral in (2.3) can be approximated by a Riemann sum of (2.1) over the interval $[0, T]$ with the partition induced by the set $\mathcal{T}$). This means that constraint (2.2) does not have to be considered in the minimization problem; if the sum of the optimal coefficients $k_1^o, \ldots, k_r^o$ is not exactly 1, they can be simply normalized by dividing each one by their sum, without altering significantly in this way the error of the fitting process.

To verify how well the continuous random variable represents the discrete one, its probability distribution

$$g_r = \mathcal{L}^{-1}[F_r(s)/s] \tag{2.4}$$

(where $\mathcal{L}^{-1}$ stands for the inverse Laplace transform) can be compared with the cumulative distribution of the discrete random variable.

The problem of finding the coefficients $k_1^o, \ldots, k_r^o$ and $a_1^o, \ldots, a_r^o$ that best fit the probability distribution implies to solve a nonlinear data fitting problem. The complexity of the problem makes practically impossible to solve it in a closed-form, thus a numerical procedure is adopted: more precisely, to solve the nonlinear minimization problem the large-scale algorithm, based on the interior-reflective Newton method [31, 32], is used, with upper and lower bounds on coefficients $k_1, \ldots, k_r$ and $a_1, \ldots, a_r$ ($0 \leq k_h \leq 1$, $a_h \geq 0$, $h = 1, \ldots, r$). The error function e_r has in general many local minima, so there is not any guarantee of reaching the global minimum starting from one particular set of initial values for the coefficients $k_1, \ldots, k_r$ and $a_1, \ldots, a_r$: the achievement of the best fitting curve depends on the algorithm chosen to compute the optimal parameters. A simple way to increase the probability of reaching the global minimum of the error function is to solve the nonlinear fitting problem for a number m of different r-tuples of initial values of the coefficients. It is

important to remark that this approach does not guarantee to find the global minimum, it just increases the probability of obtaining it.

Each r-tuple of initial values of coefficients $a_1, \ldots, a_r$ is uniformly sampled from the r-dimensional unit hypercube

$$\mathcal{A} = \{(a_1, a_2, \ldots, a_r) : 0 \leq a_h \leq 1,\, h = 1, \ldots, r\}\,,$$

while each r-tuple of initial values of coefficients $k_1, \ldots, k_r$ is uniformly sampled from the r-dimensional unit simplex

$$\mathcal{K} = \{(k_1, k_2, \ldots, k_r) : 0 \leq k_h \leq 1, \sum_{h=1}^{r} k_h = 1\}\,.$$

In particular, each r-tuple $(k_1, \ldots, k_r)$ uniformly sampled from the r-dimensional unit simplex is obtained normalizing r random samples chosen from an exponential distribution. These samples can be easily generated by using the inversion method [39], that is, by sampling uniformly from $[0, 1]$ and taking minus natural logarithm of these values. Fig. 2.2 shows an example of uniform sampling from the 3-dimensional unit simplex.

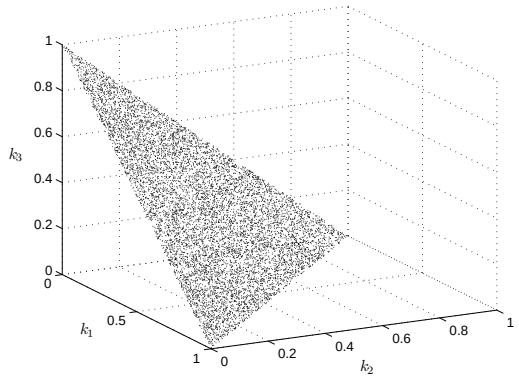

Figure 2.2: 10000 samples from the 3-dimensional unit simplex.

It is straightforward that the best fitting curve would be f_∞, that is, the function containing an infinite number of terms, and so the fitting process would never stop. However, the goal is to find the best simplest model for the

distribution, characterized by both small e_r and r; in other words, a trade-off between complexity and accuracy has to be found. Actually the process can stop when, after achieving a sufficiently small error e_r, the addition of a new term to f_r causes a negligible or even equal to zero error reduction.

In order to guarantee the not-increasing trend of e_r with respect to r, one r-tuple of initial values of coefficients $k_1, \ldots, k_r$ and $a_1, \ldots, a_r$ is appropriately chosen: in particular, referring to the function f_i containing $i > 2$ terms, $(k_1^o, \ldots, k_{i-1}^o, 0)$ and $(a_1^o, \ldots, a_{i-1}^o, 0)$ are chosen as initial values of coefficients $(k_1, \ldots, k_i)$ and $(a_1, \ldots, a_i)$ respectively, where $(k_1^o, \ldots, k_{i-1}^o)$ and $(a_1^o, \ldots, a_{i-1}^o)$ are the optimal coefficients found for the function f_{i-1}. In this way, the inequality $e_i \leq e_{i-1}$ surely holds. For $i = 1, 2$, the choice of initial values is as follows. As for function f_1, no particular initial value for coefficients k_1 and a_1 is chosen: initial values of a_1 are randomly sampled from the range $[0, 1]$, while k_1 has to be equal to one to satisfy constraint (2.2). As for f_2, one pair of initial values is chosen in the following way: $(k_1, k_2) = (1, 0)$, $(a_1, a_2) = (a_1^o, 0)$, where a_1^o is the optimal coefficient found for function f_1.

For each value of r, the procedure used to compute the optimal parameters $k_1^o, \ldots, k_r^o$ and $a_1^o, \ldots, a_r^o$ can be summarized in the following steps:

(**1**) Choose one r-tuple of initial values for parameters $k_1, \ldots, k_r$ and $a_1, \ldots, a_r$ so that the error e_r does not increase as new terms are added to the fitting function f_r (i.e., $e_{r+1} \leq e_r$):
if $r > 2$, then set
$(k_1, \ldots, k_r)^1 = (k_1^o, \ldots, k_{r-1}^o, 0)$
$(a_1, \ldots, a_r)^1 = (a_1^o, \ldots, a_{r-1}^o, 0)$
(where $k_1^o, \ldots, k_{r-1}^o$ and $a_1^o, \ldots, a_{r-1}^o$ are the optimal coefficients characterizing f_{r-1}^o)
elseif $r = 2$, then set
$(k_1, k_2)^1 = (1, 0)$
$(a_1, a_2)^1 = (a_1^o, 0)$

(where a_1^o is the optimal coefficient characterizing f_1^o)
else ($r = 1$) no particular initial value for coefficients k_1 and a_1 has to be chosen (simply choose a_1 randomly from the interval $[0, 1]$ and set $k_1 = 1$);

(**2**) Choose $m - 1$ r-tuples of initial values for parameters $k_1, \ldots, k_r$ and $a_1, \ldots, a_r$ sampling uniformly from $\mathcal{K}$ and $\mathcal{A}$ respectively:
$(k_1, \ldots, k_r)^2, \ldots, (k_1, \ldots, k_r)^m$ and $(a_1, \ldots, a_r)^2, \ldots, (a_1, \ldots, a_r)^m$;

(**3**) For each pair of r-tuples of initial values, find the best fitting curve by using the large-scale algorithm, based on the interior-reflective Newton method, with upper and lower bounds on coefficients $k_1, \ldots, k_r$ and $a_1, \ldots, a_r$ ($0 \leq k_h \leq 1$, $a_h \geq 0$, $h = 1, \ldots, r$) and compute the relative error: $e_r^1, \ldots, e_r^m$;

(**4**) Compute e_r as $min\{e_r^1, \ldots, e_r^m\}$ and hence the corresponding optimal parameters $k_1^o, \ldots, k_r^o$ and $a_1^o, \ldots, a_r^o$ characterizing f_r^o.

2.4 Case study: experimental results

A weighted random digraph containing $N = 550$ nodes and a variable number of edges, each of which is present with probability p, is considered. An edge has weight $w_1 = 1$ with probability $p_{w_1} = 0.6$, $w_2 = 2$ with probability $p_{w_2} = 0.3$ and $w_3 = 3$ with probability $p_{w_3} = 0.1$. By using the simulation tool described in Sec. 1.6, some network properties will be investigated, with p taking values in the set $\mathcal{P} = \{0.02, 0.03, \ldots, 0.1\}$. In order to perform such statistical computations on the graph, a great quantity of data needs to be gathered; to this end, for each value of p a number $q = 100$ of realizations of the graph are considered. In all these realizations, the graph turns out to be always strongly connected, i.e., there exists at least a (directed) path between every couple of nodes. The parameters characterizing the case study are summarized in Table 2.1.

Fig. 2.3 shows how global and local efficiencies vary when p changes from 0.02 to 0.1. Global efficiency is always greater than local efficiency; moreover, they are both increasing functions of p (as expected), but with different

Table 2.1: Parameters characterizing the case study.

$N = 550$
$p \in \{0.02, 0.03, \ldots, 0.1\}$
$w_1 = 1,\ w_2 = 2,\ w_3 = 3$
$p_{w_1} = 0.6,\ p_{w_2} = 0.3,\ p_{w_3} = 0.1$
$q = 100$

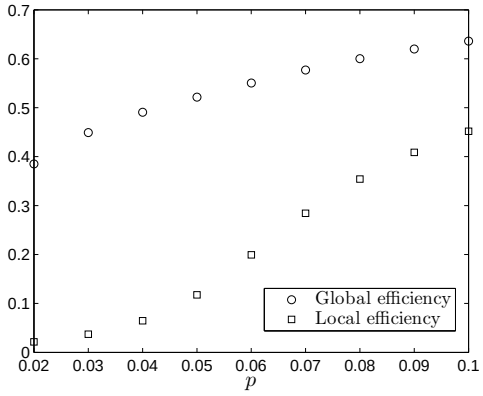

Figure 2.3: Global and local efficiencies.

rates. Global efficiency increases faster at the beginning and tends to increase slower as p gets closer to 0.1, while local efficiency seems to have three different trends, increasing slowly for small values of p, faster and almost linearly when $p \in \{0.05, \ldots, 0.07\}$ and slightly slower as p approaches 0.1. It is interesting to notice the difference between global and local efficiencies when $p \in \{0.02, \ldots, 0.04\}$: global efficiency assumes high values, even getting close to 0.5, while local efficiency increases very slowly and is always less than 0.1. The graph characterized by $p = 0.1$ achieves both high global and local efficiencies.

Fig. 2.4 shows how the diameter and cost of the graph change when p assumes values in $\mathcal{P}$. It is worthwhile to remark that, since the digraph is

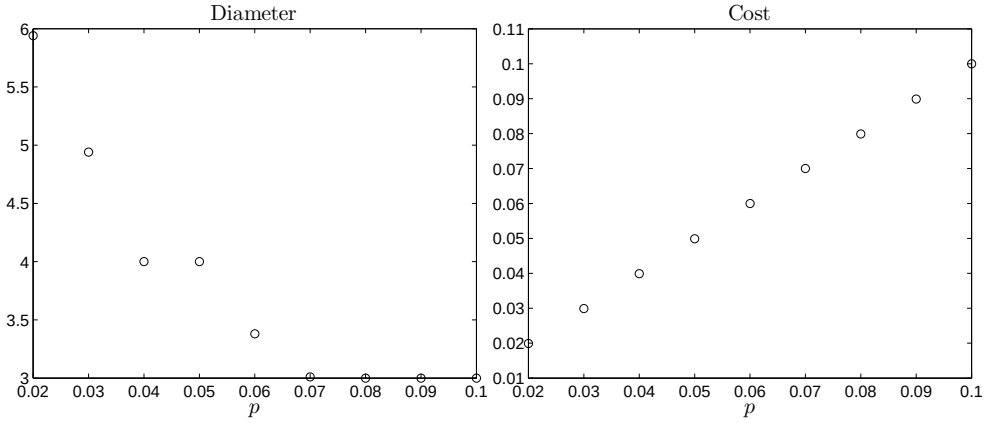

Figure 2.4: Diameter and cost.

weighted, the computation of the diameter is based on the shortest paths determined using Dijkstra's algorithm. For each value of p, the diameter is computed averaging the diameters of all the considered realizations. The diameter decreases fast when p varies from 0.02 to 0.04, and remains about unchanged from $p = 0.07$ onwards. The fact that the diameter is always less than 6 is a clear evidence of the small-world effect, that is, the existence of paths containing just few links between all pairs of vertices. As for the cost, it is always approximately equal to p: this is due to the fact that, denoting by

γ_{w_h} the cost of the edge whose weight is w_h and by $p_h = p\, p_{w_h}$ the probability to find an edge of weight w_h, Eq. (1.5) can be rewritten as

$$Cost = \frac{\sum_{h=1}^{3} p_h \gamma_{w_h}}{\sum_{h=1}^{3} p_{w_h} \gamma_{w_h}} = \frac{\sum_{h=1}^{3} p\, p_{w_h} \gamma_{w_h}}{\sum_{h=1}^{3} p_{w_h} \gamma_{w_h}} = p\,.$$

The graph characterized by $p = 0.02$ is now considered. Fig. 2.5 contains the shortest path distribution d_{sp}: the i-th element of the distribution $d_{sp}(i)$ represents the probability of finding a shortest path whose length is equal to i. It is very likely to find shortest paths of length 3 or 4, while it is very unlikely

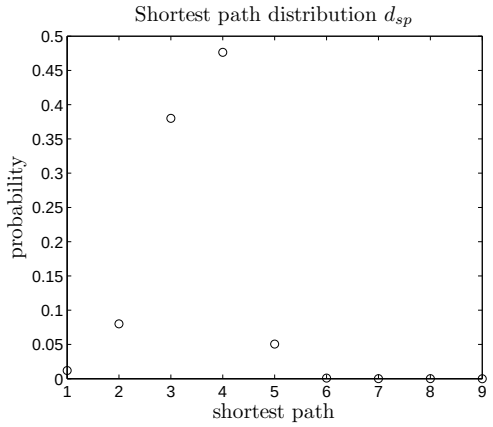

Figure 2.5: Shortest path distribution.

to find shortest paths of length greater than 5. The probability of finding shortest paths having length 9 (or even greater) turned out to be zero. This is again a clear evidence of the small-world effect: the length of the shortest path containing the maximum possible number of edges $(N-1)$, if all its edges had unit weight (i.e., in the better case), would be $N - 1 = 549$, much greater than the maximum length (8) found in the simulations.

Fig. 2.6 contains the edge load distribution d_{el} and the node load distribution d_{nl}: $d_{el}(i)$ ($d_{nl}(i)$) is the probability of finding an edge (node) whose load is i. Unlike the node load distribution, the edge load distribution is a multimodal distribution, having more than one peak. Both d_{el} and d_{nl} could

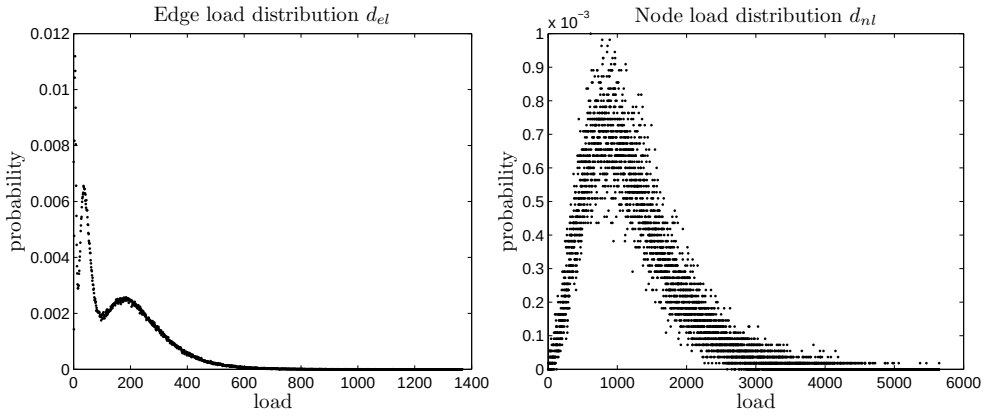

Figure 2.6: Edge and node load distributions.

be used in congestion control problems, giving information about the traffic intensity on edges and nodes of the network.

The techniques explained in Sec. 2.3 are now used to find the best fitting curves for d_{el} and d_{nl}, showing how error e_r decreases or at least does not increase as new terms are added to f_r, i.e., how e_r is a not-increasing function of r. For each value of r, a number $m = 100$ of r-tuples of initial values for coefficients $k_1, \ldots, k_r$ and $a_1, \ldots, a_r$ are considered.

Fig. 2.7 compares distribution d_{el} with the best fitting curves $f_1^o, \ldots, f_8^o$; the probability to find an edge whose load is greater than $T = 1367$ is zero. It is easy to see how the fitting function f_r^o better approximates the distribution d_{el} as the number of terms r increases. In particular, if $r > 5$ it is actually difficult to distinguish between original and fitted data, and this is confirmed by Fig. 2.8: the fitting error e_6 is very small and slightly differs from e_7 and e_8, thus function f_6^o could be a simple and quite good fitting model for distribution d_{el}.

As for distribution d_{nl}, Fig. 2.9 shows the comparison between original and fitted data. The independent discrete random variable, that is, the node load, assumes values in the range $[0, T]$, where $T = 5647$. It is again evident how f_r^o gets closer to the distribution d_{nl} as the number of terms r increases. Fig. 2.8

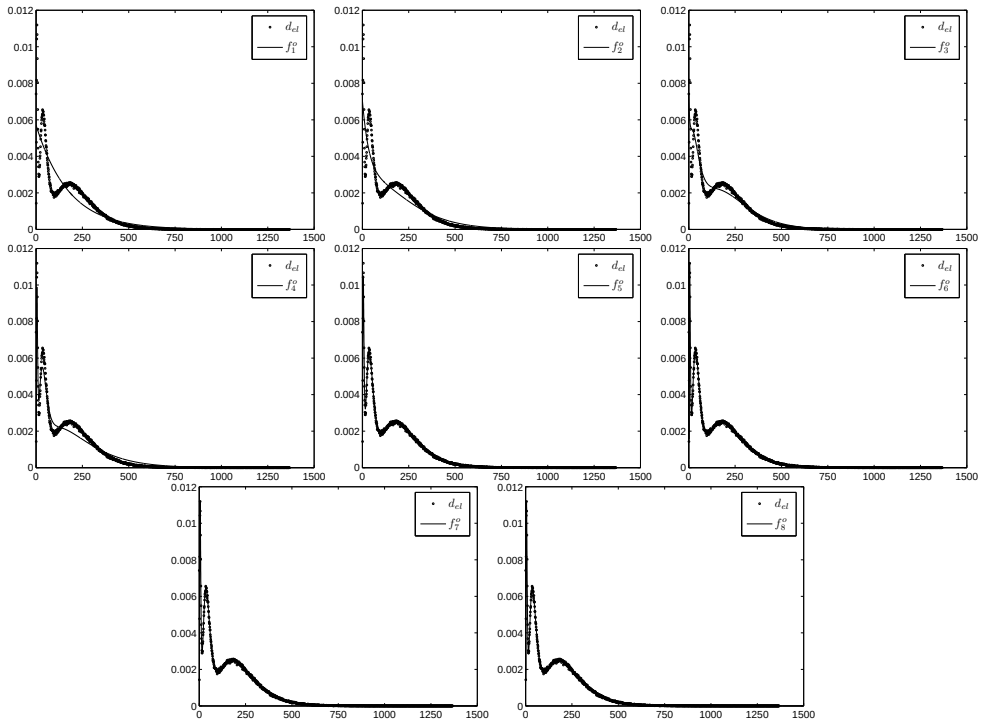

Figure 2.7: Comparison between d_{el} and fitting functions $f_1^o, \ldots, f_8^o$.

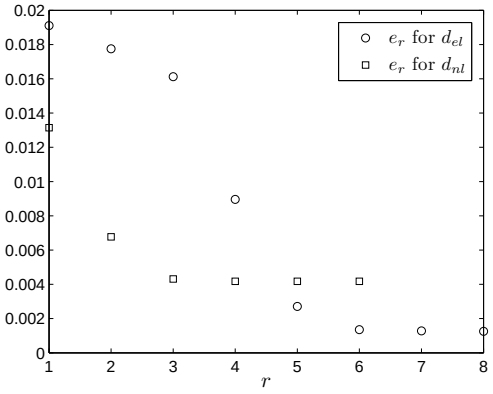

Figure 2.8: Fitting error e_r for d_{el} $(r = 1, \ldots, 8)$ and d_{nl} $(r = 1, \ldots, 6)$.

showing fitting error e_r seems to suggest f_3^o as a simple and quite good fitting model for distribution d_{nl}.

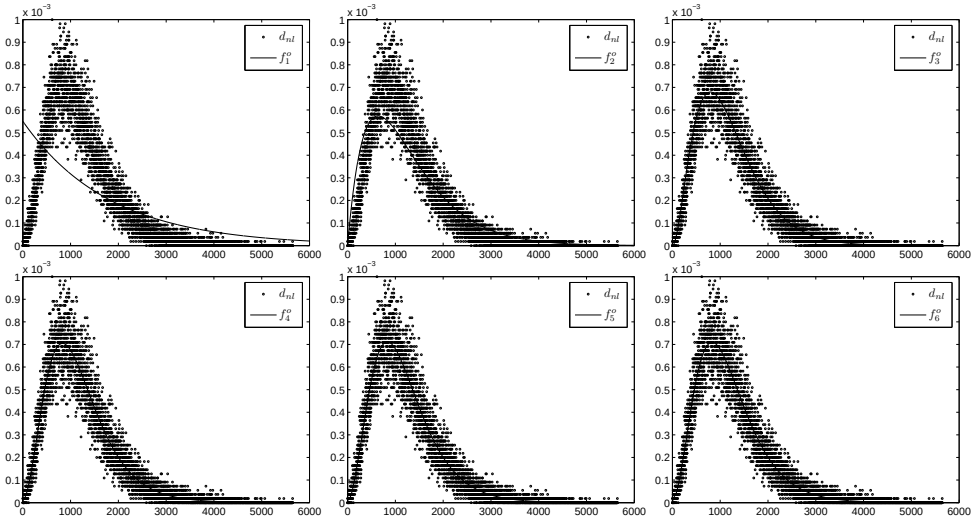

Figure 2.9: Comparison between d_{nl} and fitting functions $f_1^o, \ldots, f_6^o$.

With reference to d_{nl}, in order to assess how well the continuous random variable represents the discrete one, Fig. 2.10 shows the comparison between the distribution g_3^o (Eq. (2.4))

$$\begin{aligned} g_3^o &= \mathcal{L}^{-1}[F_r^o(s)/s] = \\ &= k_1^o(1 - e^{-a_1^o t}) + k_2^o(1 - (1 + a_2^o t)e^{-a_2^o t}) + \\ &+ \frac{k_3^o}{2}(2 - ((a_3^o)^2 t^2 + 2 + 2a_3^o t)e^{-a_3^o t}) \end{aligned}$$

and the cumulative distribution D_{nl} of the discrete random variable ($D_{nl}(i)$ is the probability of finding a node whose load is less than or equal to i). The two distributions are very close to each other, thus the continuous random variable effectively represents the discrete one.

Finally, with reference to both d_{el} and d_{nl}, Fig. 2.11 shows how parameter a_1^o of f_1^o changes when p varies from 0.02 to 0.1. By fitting the parameter values, it is possible to get a_1^o for all the values of p in the range $[0.02, 0.1]$.

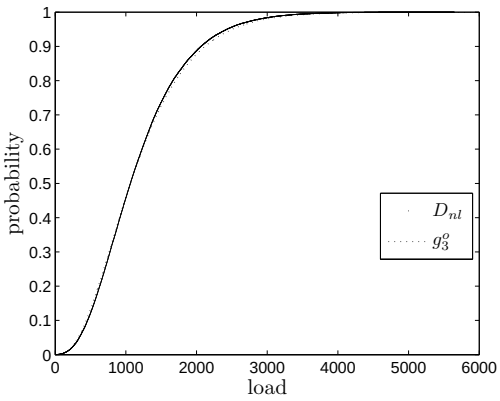

Figure 2.10: Comparison between the distribution g_3^o and the cumulative distribution D_{nl}.

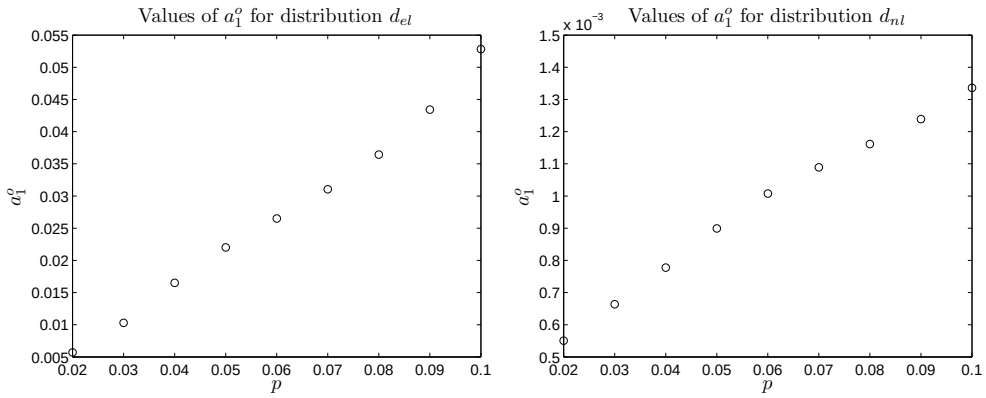

Figure 2.11: Values of parameter a_1^o relative to values of p in $\mathcal{P}$, for distributions d_{el} and d_{nl}.

2.5 Conclusions

A weighted directed random graph was used as network model. Some properties, like global and local efficiency, diameter and cost of the graph were computed and their reliance on probability p was investigated. A fitting procedure based on the method of stages was presented and used to find the probability density function that best fits an empirical distribution. An example showing the effectiveness of the proposed method and the way as optimal parameters rely on probability p was also illustrated. The general structure of the proposed fitting technique allows to model several aspects of complex networks and makes possible its use in many different fields.

Chapter 3

Tracking control with time delay compensation for linear time invariant networked control systems

The chapter deals with the tracking control problem of linear time invariant (LTI) systems when the plant and the controller belong to the same network. Time delays can degrade significantly the performance of a networked control system, eventually leading to instability. The problem characterized by constant and known network delays is analytically examined, showing how to construct a plant state predictor in order to compensate the time delays between the plant and the controller, so to allow the tracking of a reference signal. Computer simulations illustrate the effectiveness of the proposed technique, also when time delays slightly vary around a mean value. The content of this chapter is based on [17].

3.1 Introduction

Networked control systems (NCSs) are control systems whose control loop is closed around a communication network. The systems are thus nodes of the same network and all the information they exchange is subject to time delays induced by the shared medium. Examples of NCSs can be found in several fields, like for example automotive industry, teleautonomy, teleoperation, robot formation and automated manufacturing systems [7]. The main advantages of an NCS are reduced system wiring, ease of system diagnosis and maintenance, and increased system agility. The drawback is the augmented complexity of analysis and design of NCSs with respect to classical feedback control systems [131].

The network-induced delay (plant-to-controller delay and controller-to-plant delay), either constant or time-varying, can degrade the performance of control systems designed without considering the delay and can even destabilize the system [131]. Research involving distributed control over networks has primarily focused on the development of protocols for network communications and on the design and analysis of network controllers, mostly utilizing wired networks [100]. Two network types, token-passing bus (ControlNet) and CAN bus (DeviceNet), are specifically designed as control networks, that is, for frequent transmission of small data packets and guaranteed times of arrival [77]. Another network type, Ethernet, is instead designed as a data network for less frequent transmissions of large data packets at high data rates [77, 111]. General data networks are rapidly advancing to be the networks of choices for many applications due to their flexibility and lower costs [27]. Recent control methodologies for a closed-loop control system over a data network are described in [117]. Wireless Ethernet (802.11) solutions have also been investigated as control networks [79, 128, 129].

Several papers have focused on stability analysis of NCSs. In [122] the network separates the sensor from the controller; stability is provided for two scheduling methods, including try once discard (TOD) and a statically sched-

uled arrangement. The authors introduce the notion of maximum allowable transfer interval (MATI), and find that interval value under which stability is guaranteed. In [131] the network is placed between the sensor and controller and also between the controller and actuator; the allowable delay is obtained analytically in the simplest case, while for more complex situations the range of stable delays is found through simulation. Beyond the pure analysis of stability and performance of existing controllers, some work has focused on the design of network controllers to address the expected communication delays [22, 80].

The observer design used in this chapter is mainly inspired by the work of Olbrot [96], in which it is shown that the observer problem for general systems with delays can be reformulated so as to get an easier case without output delays.

3.2 Problem statement and solution

A plant P and a controller G are linear time invariant (LTI) systems representing two nodes of the same network (Fig. 3.1). All signals they exchange are

Figure 3.1: Networked control system.

therefore subject to time delays: precisely, the *plant-to-controller* time delay τ_{pg} and the *controller-to-plant* time delay τ_{gp}. Both delays are assumed to be known and constant, but not necessarily equal. The plant model has the following state-space representation:

$$\text{Plant}: \begin{cases} \dot{x}(t) = A\,x(t) + B\,u(t)\,, \\ y(t) = C\,x(t)\,, \end{cases} \tag{3.1}$$

where $x \in \mathbb{R}^n$, $u, y \in \mathbb{R}$ and A, B, C are real matrices of appropriate dimensions. Due to the presence of the network, the plant output $y(t)$ does not

coincide with the controller input (Fig. 3.2), which is given by

$$u_g(t) = C\,x(t - \tau_{pg})\,. \tag{3.2}$$

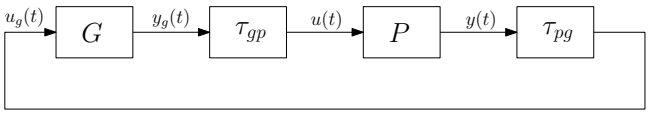

Figure 3.2: Block scheme of the networked control system shown in Fig. 3.1.

Problem 1. *Design of an output feedback control law such that the eigenvalues of the closed-loop system have negative real part and the plant output $y(t)$ asymptotically tracks a given reference signal $v(t)$, which is possibly generated by a linear time invariant exosystem (with the eigenvalues having nonnegative real part).*

Without loss of generality, it is assumed that the zeros of the plant transfer function and the eigenvalues of the exosystem dynamic matrix do not have values in common.

If the pair (C, A) is detectable, it is possible to design an asymptotic plant state observer whose dynamics is

$$\begin{cases} \dot{\hat{x}}(t) = (A - L\,\tilde{C})\,\hat{x}(t) + B\,u(t) + L\,\tilde{y}(t)\,, \\ \tilde{y}(t) = u_g(t) + \tilde{C}\,z(t)\,, \\ \dot{z}(t) = A\,z(t) + B\,u(t) - e^{A\tau_{pg}}\,B\,u(t - \tau_{pg})\,, \\ z(0) = 0\,, \end{cases} \tag{3.3}$$

where $\hat{x}(t)$ is the plant state estimate, $\tilde{C} = C\,e^{-A\tau_{pg}}$ and gain matrix L is such that the estimation error $\tilde{x}(t) = x(t) - \hat{x}(t)$ asymptotically converges to zero.

The tracking task is accomplished by introducing the internal model, whose equations are

$$\begin{cases} \dot{x}_m(t) = A_m\,x_m(t) + B_m\,u_m(t)\,, \\ u_m(t) = v(t + \tau_{gp}) - C\,\check{x}(t)\,, \\ y_m(t) = x_m(t)\,, \end{cases} \tag{3.4}$$

where $\check{x}(t)$ is the plant state prediction at time $t + \tau_{gp}$:

$$\begin{cases} \check{x}(t) = e^{A\tau_{gp}} \left(\hat{x}(t) + w(t)\right), \\ \dot{w}(t) = A\, w(t) - B\, u(t) + e^{-A\tau_{gp}}\, B\, u(t + \tau_{gp}), \\ w(0) = 0\,. \end{cases} \tag{3.5}$$

If the pair

$$\left(\begin{bmatrix} A & 0 \\ -B_m\, C & A_m \end{bmatrix}, \begin{bmatrix} B \\ 0 \end{bmatrix} \right) \tag{3.6}$$

is stabilizable, it is possible to find a gain matrix $K = [K_1\ K_2]$ such that the controller output

$$y_g(t) = K_1\, \check{x}(t) + K_2\, x_m(t) \tag{3.7}$$

ensures that all the eigenvalues of the closed-loop system

$$\begin{cases} \dot{x}(t) = (A + B\, K_1)\, x(t) + B\, K_2\, \bar{x}_m(t) - B\, K_1\, e^{A\tau_{gp}}\, e^{-(A - L\tilde{C})\tau_{gp}}\, \tilde{x}(t), \\ \dot{\bar{x}}_m(t) = -B_m\, C\, x(t) + A_m\, \bar{x}_m(t) + B_m\, C\, e^{A\tau_{gp}}\, e^{-(A - L\tilde{C})\tau_{gp}}\, \tilde{x}(t) + B_m\, v(t), \\ \dot{\tilde{x}}(t) = (A - L\,\tilde{C})\, \tilde{x}(t), \end{cases} \tag{3.8}$$

where $\bar{x}_m(t) = x_m(t - \tau_{gp})$, have negative real part. The controller G is completely characterized by (3.2)–(3.5) and (3.7), and the relative block scheme is shown in Fig 3.3. The closed-loop dynamic matrix is an upper block triangular matrix, whose diagonal blocks are the closed-loop dynamic matrix of the control scheme of Fig. 3.4 and $A - L\,\tilde{C}$. Gain matrices K and L can therefore be independently designed, and gain matrix K can be designed as if the plant state were fully measurable and both time delays τ_{pg} and τ_{gp} were not present.

Theorem 1. *Assuming that the zeros of the plant transfer function and the eigenvalues of the exosystem dynamic matrix do not have values in common, necessary and sufficient conditions for the design of an output feedback control law such that the eigenvalues of the closed-loop system have negative real part and the plant output $y(t)$ asymptotically tracks the reference signal $v(t)$ are the stabilizability of the pair* (3.6) *and the detectability of the pair* (C, A).

Proof. Putting together the plant dynamics and the time delay τ_{pg}, the following equations are obtained:

$$\text{Plant \& delay } \tau_{pg}: \begin{cases} \dot{x}(t) = A\,x(t) + B\,u(t)\,, \\ \bar{y}(t) = C\,x(t-\tau_{pg})\,. \end{cases} \tag{3.9}$$

Taking $t = 0$ as the initial time, the solution of the first equation of (3.1) is:

$$x(t) = e^{At}\,x(0) + \int_0^t e^{A(t-\tau)}\,B\,u(\tau)\,\mathrm{d}\tau\,; \tag{3.10}$$

hence

$$x(t-\tau_{pg}) = e^{-A\tau_{pg}}\,e^{At}\,x(0) + \int_0^{t-\tau_{pg}} e^{A(t-\tau_{pg}-\tau)}\,B\,u(\tau)\,\mathrm{d}\tau\,. \tag{3.11}$$

Substituting the expression of $e^{At}\,x(0)$ obtainable from (3.10) into (3.11),

$$x(t-\tau_{pg}) = e^{-A\tau_{pg}}\,(x(t) - z(t))\,, \tag{3.12}$$

where

$$z(t) = \int_{t-\tau_{pg}}^{t} e^{A(t-\tau)}\,B\,u(\tau)\,\mathrm{d}\tau\,. \tag{3.13}$$

Assuming that $u(t) = 0$, $\forall\, t < 0$, the dynamics of the new variable $z(t)$ can be easily derived from (3.13):

$$\begin{cases} \dot{z}(t) = A\,z(t) + B\,u(t) - e^{A\tau_{pg}}\,B\,u(t-\tau_{pg})\,, \\ z(0) = 0\,. \end{cases}$$

The second equation of (3.9) can thus be rewritten as

$$\bar{y}(t) = C\,e^{-A\tau_{pg}}\,(x(t) - z(t)) = \tilde{C}\,(x(t) - z(t))\,,$$

where $\tilde{C} = C\,e^{-A\tau_{pg}}$. Accordingly, system (3.9) becomes:

$$\text{Plant \& delay } \tau_{pg}: \begin{cases} \dot{x}(t) = A\,x(t) + B\,u(t)\,, \\ \tilde{y}(t) = \tilde{C}\,x(t)\,, \end{cases} \tag{3.14}$$

where $\tilde{y}(t) = \bar{y}(t) + \tilde{C}\, z(t)$. On the base of (3.14) and under the assumption that the pair $(\tilde{C}, A)$ is detectable, it is well known how to design an asymptotic plant state observer such that the estimation error $\tilde{x}(t)$ asymptotically converges to zero. Notice that the pair $(\tilde{C}, A)$ is detectable if and only if the pair (C, A) is detectable. In fact,

$$\begin{bmatrix} \lambda I - A \\ \hdashline \tilde{C} \end{bmatrix} \underbrace{e^{A\tau_{pg}}}_{\text{full rank}} = \begin{bmatrix} (\lambda I - A)\, e^{A\tau_{pg}} \\ \hdashline C \end{bmatrix} = \begin{bmatrix} e^{A\tau_{pg}}\, (\lambda I - A) \\ \hdashline C \end{bmatrix}$$

and

$$\underbrace{\left[\begin{array}{c:c} e^{-A\tau_{pg}} & 0 \\ \hdashline 0 & 1 \end{array}\right]}_{\text{full rank}} \begin{bmatrix} e^{A\tau_{pg}}\, (\lambda I - A) \\ \hdashline C \end{bmatrix} = \begin{bmatrix} \lambda I - A \\ \hdashline C \end{bmatrix},$$

hence

$$\operatorname{rank} \begin{bmatrix} \lambda I - A \\ \hdashline \tilde{C} \end{bmatrix} = \operatorname{rank} \begin{bmatrix} \lambda I - A \\ \hdashline C \end{bmatrix}.$$

By choosing for the plant state observer the dynamics

$$\dot{\hat{x}}(t) = (A - L\,\tilde{C})\,\hat{x}(t) + B\,u(t) + L\,\tilde{y}(t)\,,$$

where the gain matrix L is such that $\lambda(A - L\,\tilde{C}) \subseteq \mathbb{C}^-$, the following dynamics for the estimation error is obtained:

$$\begin{aligned} \dot{\tilde{x}}(t) &= \dot{x}(t) - \dot{\hat{x}}(t) \\ &= A\,x(t) + B\,u(t) - (A - L\,\tilde{C})\,\hat{x}(t) - B\,u(t) - L\,\tilde{C}\,x(t) \\ &= (A - L\,\tilde{C})\,\tilde{x}(t)\,, \end{aligned} \tag{3.15}$$

i.e.,

$$\tilde{x}(t) = e^{(A - L\tilde{C})t}\,\tilde{x}(0)\,.$$

The network also introduces the controller-to-plant time delay τ_{gp}, thus a further delay compensation action needs to be implemented. In other words, the controller output must contain the plant state prediction instead of the plant state estimate.

With reasonings similar to those used to get Eq. (3.12), it is straightforward to see that

$$x(t+\tau_{gp}) = e^{A\tau_{gp}}\,(x(t)+w(t))\,, \tag{3.16}$$

where

$$w(t) = \int_t^{t+\tau_{gp}} e^{A(t-\tau)}\,B\,u(\tau)\,\mathrm{d}\tau\,. \tag{3.17}$$

Similarly to $z(t)$, the dynamics of $w(t)$ can be obtained from (3.17): since $u(t)=0$, $\forall\, t<\tau_{gp}$,

$$\begin{cases} \dot{w}(t) = A\,w(t) - B\,u(t) + e^{-A\tau_{gp}}\,B\,u(t+\tau_{gp})\,, \\ w(0) = 0\,. \end{cases}$$

If the plant state at time t were known, by means of (3.16) it would be possible to compute the plant state at time $t+\tau_{gp}$. The right-hand side of (3.16) with the replacement of $x(t)$ with $\hat{x}(t)$ becomes

$$\begin{aligned} e^{A\tau_{gp}}\,(\hat{x}(t)+w(t)) &= e^{A\tau_{gp}}\,(x(t)-\tilde{x}(t)+w(t)) \\ &= e^{A\tau_{gp}}\,(x(t)+w(t)) - e^{A\tau_{gp}}\,\tilde{x}(t) \\ &= x(t+\tau_{gp}) - e^{A\tau_{gp}}\,e^{-(A-L\tilde{C})\tau_{gp}}\,\tilde{x}(t+\tau_{gp}) =: \check{x}(t)\,, \end{aligned}$$

which approximates the plant state at time $t+\tau_{gp}$ as soon as the estimation error approaches zero: $\check{x}(t)$ is then an asymptotic plant state prediction at time $t+\tau_{gp}$.

If the plant does not contain the model of the reference signal, the controller has to contain the internal model equations (3.4), which guarantee that the plant output $y(t)$ asymptotically tracks the reference signal $v(t)$. The gain matrices K_1, K_2 appearing in the controller output (3.7) have to be chosen in order to guarantee that all the eigenvalues of the closed-loop system have negative real part. The plant input is simply the controller output delayed by τ_{gp}:

$$u(t) = y_g(t-\tau_{gp}) = K_1\,\check{x}(t-\tau_{gp}) + K_2\,x_m(t-\tau_{gp})\,.$$

The controller block scheme is shown in Fig. 3.3.

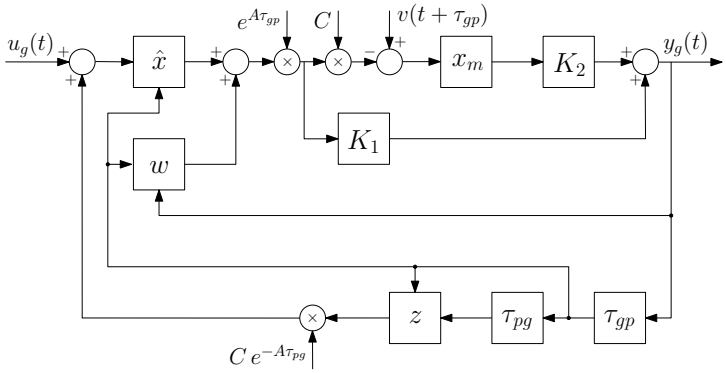

Figure 3.3: Block scheme of the controller G.

The closed-loop dynamics of $x(t)$ is

$$\begin{aligned}\dot{x}(t) &= A\,x(t) + B\,u(t)\\ &= A\,x(t) + B\,K_1\,\breve{x}(t-\tau_{gp}) + B\,K_2\,x_m(t-\tau_{gp})\\ &= (A+B\,K_1)\,x(t) - B\,K_1\,e^{A\tau_{gp}}\,e^{-(A-L\tilde{C})\tau_{gp}}\tilde{x}(t) + B\,K_2\,\bar{x}_m(t)\,,\end{aligned} \tag{3.18}$$

where $\bar{x}_m(t) := x_m(t-\tau_{gp})$ and its closed-loop dynamics is

$$\begin{aligned}\dot{\bar{x}}_m(t) &= A_m\,\bar{x}_m(t) + B_m\,u_m(t-\tau_{gp})\\ &= A_m\,\bar{x}_m(t) + B_m\,v(t) - B_m\,C\,\breve{x}(t-\tau_{gp})\\ &= A_m\,\bar{x}_m(t) + B_m\,v(t) - B_m\,C\,x(t) + B_m\,C\,e^{A\tau_{gp}}\,e^{-(A-L\tilde{C})\tau_{gp}}\,\tilde{x}(t)\,.\end{aligned} \tag{3.19}$$

Equations (3.18), (3.19) and (3.15) characterize the closed-loop system (3.8), and can be rewritten in the following compact form:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{\bar{x}}_m(t) \\ \dot{\tilde{x}}(t) \end{bmatrix} = A_{cl} \begin{bmatrix} x(t) \\ \bar{x}_m(t) \\ \tilde{x}(t) \end{bmatrix} + B_{cl}\,v(t)\,,$$

where

$$A_{cl} = \left[\begin{array}{cc:c} A+B\,K_1 & B\,K_2 & -B\,K_1\,e^{A\tau_{gp}}\,e^{-(A-L\tilde{C})\tau_{gp}} \\ -B_m\,C & A_m & B_m\,C\,e^{A\tau_{gp}}\,e^{-(A-L\tilde{C})\tau_{gp}} \\ \hdashline 0 & 0 & A-L\,\tilde{C} \end{array}\right],\; B_{cl} = \left[\begin{array}{c} 0 \\ B_m \\ \hdashline 0 \end{array}\right].$$

The gain matrix $K = [K_1\ K_2]$ has to be chosen such that $\lambda(A_{cl}) \subseteq \mathbb{C}^-$, i.e., such that all the eigenvalues of the matrix in the upper-left corner of A_{cl},

$$F := \begin{bmatrix} A + B\,K_1 & B\,K_2 \\ -B_m\,C & A_m \end{bmatrix},$$

lie in the open left complex half-plane: $\lambda(F) \subseteq \mathbb{C}^-$. Summarizing, the necessary and sufficient conditions for the closed-loop system (3.8) having all the eigenvalues with negative real part are the stabilizability of the pair (3.6) and the detectability of the pair (C, A).

It is interesting to notice that matrix F coincides with the closed-loop dynamic matrix obtained ignoring network delays (both τ_{pg} and τ_{gp}) and assuming the plant state is fully measurable. In fact, in this case the equations of the closed-loop system are

$$\begin{cases} \dot{x}(t) = A\,x(t) + B\,u(t)\,, \\ y(t) = C\,x(t)\,, \\ u(t) = K_1\,x(t) + K_2\,y_m(t)\,, \\ \dot{x}_m(t) = A_m\,x_m(t) + B_m\,u_m(t)\,, \\ y_m(t) = x_m(t)\,, \\ u_m(t) = v(t) - y(t)\,, \end{cases} \tag{3.20}$$

or, in a more compact form,

$$\begin{bmatrix} \dot{x}(t) \\ \dot{x}_m(t) \end{bmatrix} = \overbrace{\begin{bmatrix} A + B\,K_1 & B\,K_2 \\ -B_m\,C & A_m \end{bmatrix}}^{F} \begin{bmatrix} x(t) \\ x_m(t) \end{bmatrix} + \begin{bmatrix} 0 \\ B_m \end{bmatrix} v(t)\,. \tag{3.21}$$

The corresponding block scheme is shown in Fig. 3.4. This means that, even in presence of constant and known time delays between the plant and the controller, it is possible to *separate* the plant state observer design problem from the control law design problem. Gain matrix K can be designed assuming that the plant state is fully measurable and both time delays τ_{pg} and τ_{gp} are

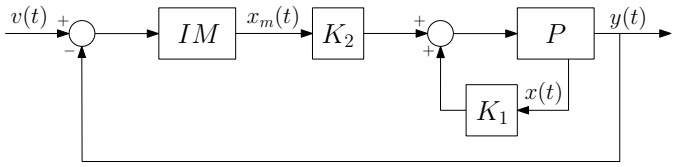

Figure 3.4: Block scheme relative to equations (3.20).

not present. The eigenvalues of the closed-loop dynamic matrix A_{cl} are those of $A - L\,\tilde{C}$ and those of matrix F:

$$\lambda(A_{cl}) = \lambda(F) \cup \lambda(A - L\,\tilde{C})\,.$$

It remains to show that the plant output $y(t)$ asymptotically tracks the reference signal $v(t)$, i.e.,

$$\lim_{t\to\infty} e(t) = 0\,, \tag{3.22}$$

where $e(t) = v(t) - y(t)$ is the tracking error. If $\tilde{x}(t) = 0$, the equations (3.8) describing the closed-loop system become

$$\begin{cases} \dot{x}(t) = (A + B\,K_1)\,x(t) + B\,K_2\,\bar{x}_m(t)\,, \\ \dot{\bar{x}}_m(t) = -B_m\,C\,x(t) + A_m\,\bar{x}_m(t) + B_m\,v(t)\,, \\ \dot{\tilde{x}}(t) = 0\,, \end{cases}$$

thus reducing to (3.21) with $\bar{x}_m(t)$ in place of $x_m(t)$. From classical results of control theory, it directly follows that (3.22) holds. □

3.3 Numerical results

An LTI plant P characterized by the following matrices is considered:

$$A = \begin{bmatrix} 0 & -5 & 0 & -4 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \; B = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \; C = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}.$$

The eigenvalues of A are the roots of the polynomial $s^4 + 5\,s^2 + 4$:

$$\lambda(A) = \{\pm\, i, \pm\, 2\, i\}\,.$$

The initial state of the plant is $x(0) = [1\ 0\ 3\ 6]^T$. The reference signal is a ramp, $v(t) = t$; since the plant does not include the model of the reference signal, the controller has to contain the internal model equations, with matrices

$$A_m = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix},\ B_m = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

The initial state of the internal model is chosen equal to zero: $x_m(0) = [0\ 0]^T$.

The pair (3.6) and the pair (C, A) are controllable and observable respectively. Therefore, it is possible to design an output feedback control law such that all the eigenvalues of the closed-loop system have negative real part and the plant output asymptotically tracks the reference signal. The computation of gain matrix L is based on the algorithm from [66], while gain matrix K is calculated by LQR technique. In particular, gain matrices L and K are chosen such that

$$\lambda(A - L\,\tilde{C}) = \{-11, -12, -13, -14\}\,,$$
$$\lambda(F) = \{-1.9079 \pm 2.7167\, i\,,\ -0.9963 \pm 0.2786\, i\,,\ -0.2868 \pm 0.5068\, i\}\,.$$

As for the time delays, three different scenarios are considered. In the first scenario (S_1), both delays τ_{pg} and τ_{gp} are assumed to be constant and known: $\tau_{pg} = 100\,[ms]$, $\tau_{gp} = 150\,[ms]$. The plant state estimation error $\tilde{x}(t)$ is shown in Fig. 3.5, while Fig. 3.6 illustrates the plant state prediction error $x(t) - \check{x}(t)$. Obviously, the plant state prediction error is meaningful from $t = \tau_{gp}$ onwards. Fig. 3.7 shows the plant output $y(t)$ versus the reference signal $v(t)$ (dashed line) and the corresponding tracking error $e(t)$. Finally, Fig. 3.8 illustrates the control signal $u(t)$. The shown figures support the effectiveness of the proposed delay compensation technique.

In the second scenario (S_2), τ_{pg} and τ_{gp} are normally distributed random variables with mean $\mu_{\tau_{pg}} = 100\,[ms]$ and $\mu_{\tau_{gp}} = 150\,[ms]$ respectively; both

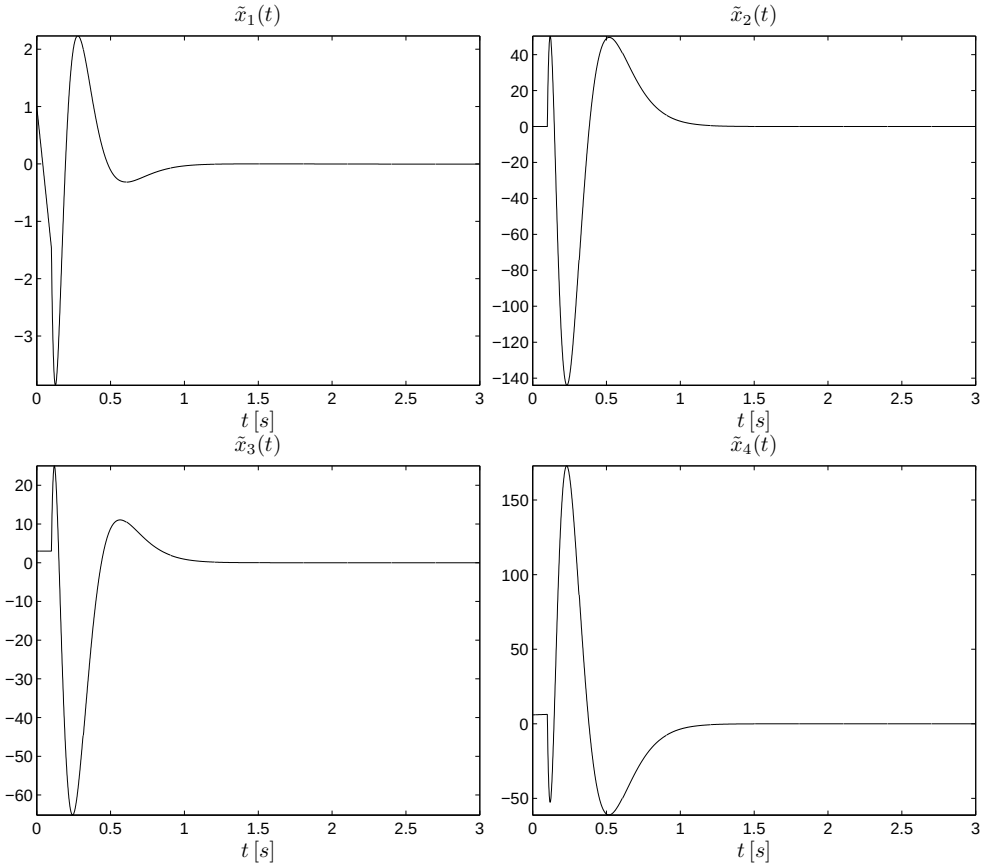

Figure 3.5: S_1: Plant state estimation error $\tilde{x}(t)$.

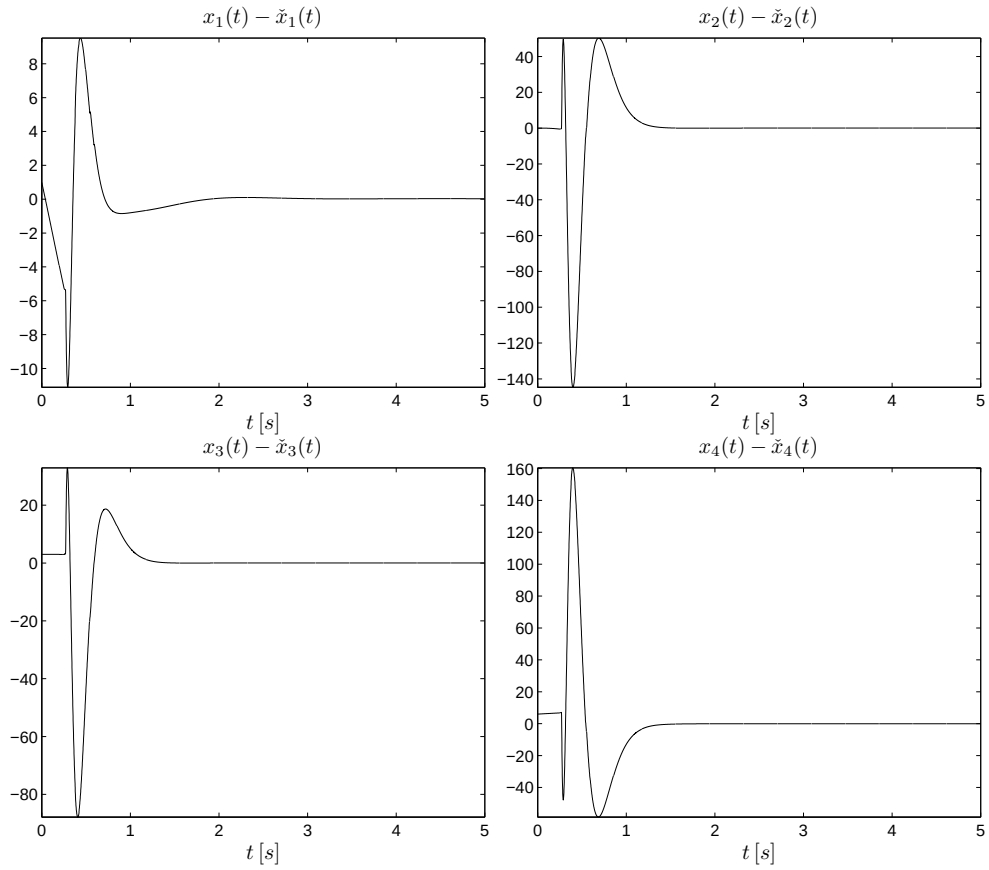

Figure 3.6: S_1: Plant state prediction error $x(t) - \check{x}(t)$.

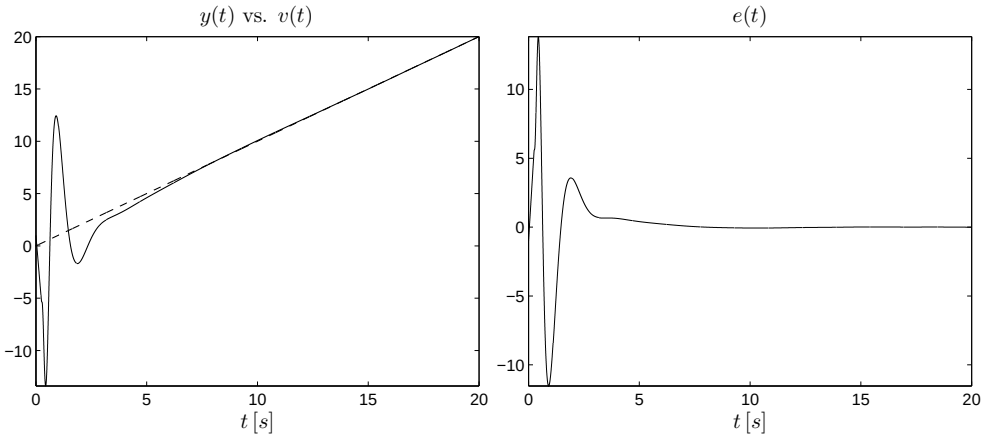

Figure 3.7: S_1: Plant output $y(t)$ versus the reference signal $v(t)$ (dashed line) and the corresponding tracking error $e(t)$.

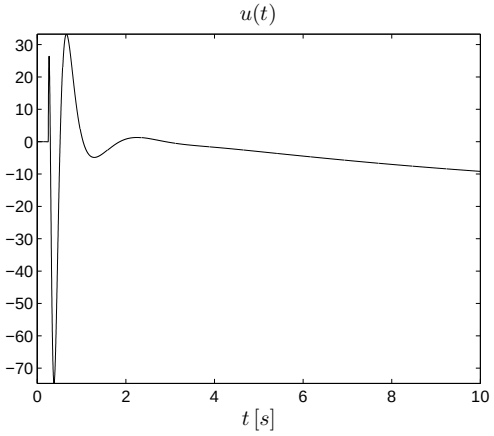

Figure 3.8: S_1: Control signal $u(t)$.

time delays have standard deviation $\sigma = 4\,[ms]$ and change every $5\,[s]$. The delay compensation procedure is accomplished approximating the actual time delays with their mean values. Fig. 3.9 shows the plant state estimation error $\tilde{x}(t)$: compared with Fig. 3.5, the estimation error presents a longer con-

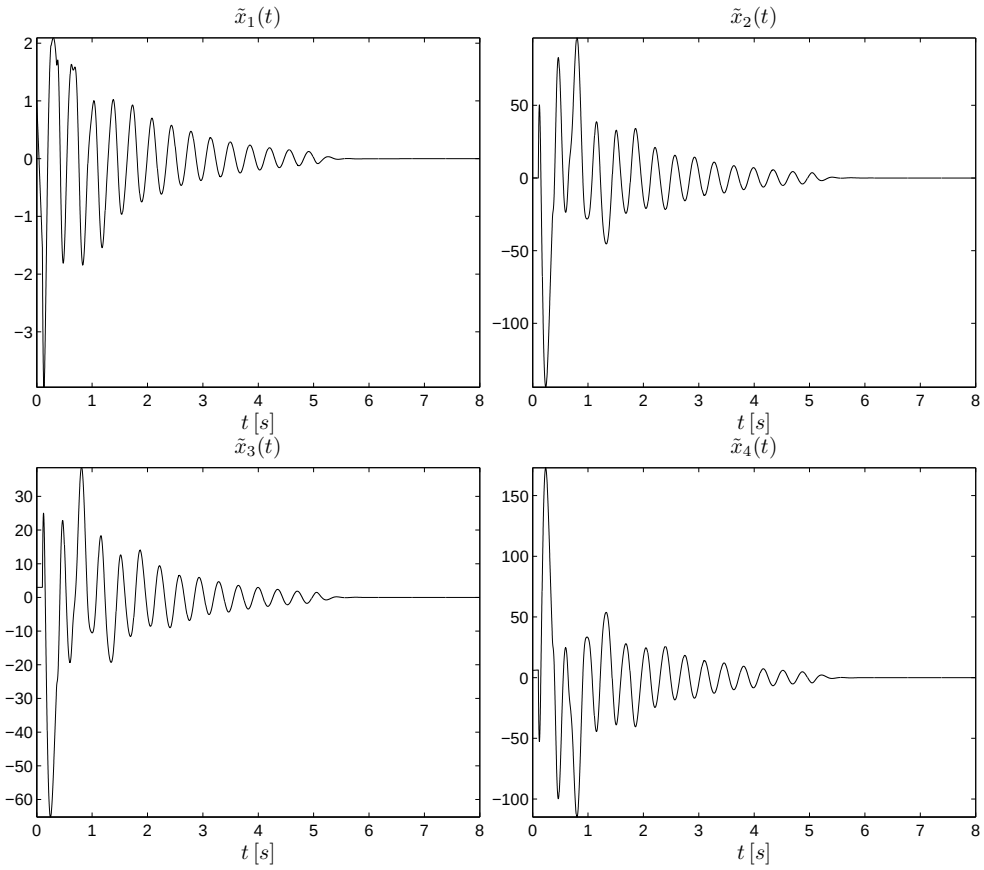

Figure 3.9: S_2: Plant state estimation error $\tilde{x}(t)$.

vergence time characterized by oscillations whose amplitude approaches zero after about $7\,[s]$. Fig. 3.10 contains the plant state prediction error $x(t) - \breve{x}(t)$: similarly to the estimation error, the oscillations become negligible after about $7\,[s]$. Fig. 3.11 illustrates the plant output $y(t)$ versus the reference signal $v(t)$ (dashed line) and the corresponding tracking error $e(t)$: the tracking error is

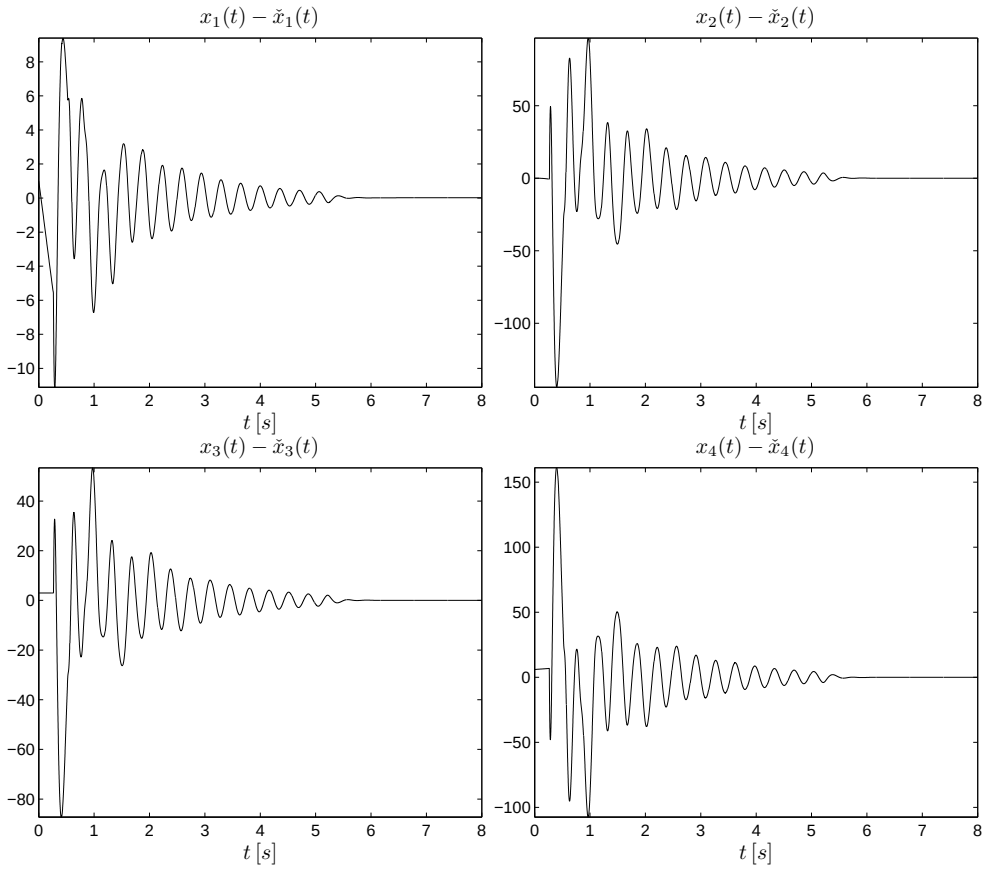

Figure 3.10: S_2: Plant state prediction error $x(t) - \check{x}(t)$.

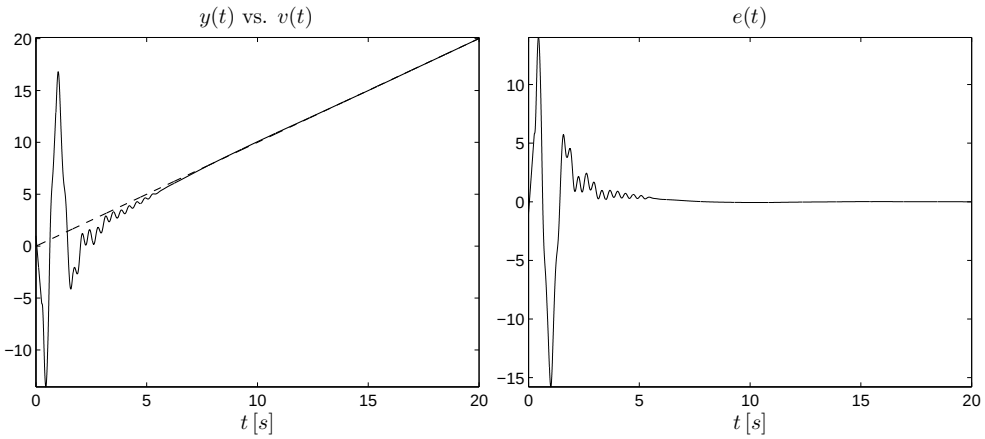

Figure 3.11: S_2: Plant output $y(t)$ versus the reference signal $v(t)$ (dashed line) and the corresponding tracking error $e(t)$.

almost zero from $t = 10\,[s]$ onwards. Finally, Fig. 3.12 shows the control signal $u(t)$: with respect to the first scenario, the control signal presents oscillations

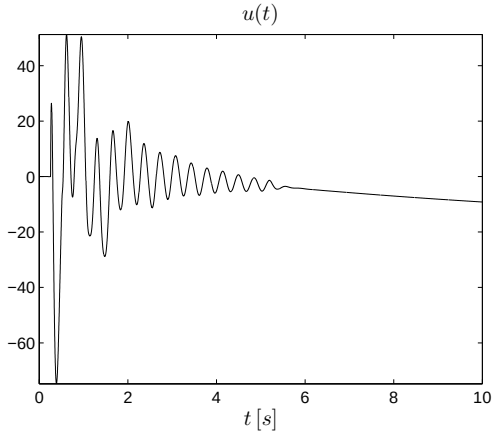

Figure 3.12: S_2: Control signal $u(t)$.

that become negligible from $t = 6\,[s]$ onwards. The computer simulations show that the proposed delay compensation technique also works in the second scenario, in which time delays slightly vary around their mean values.

In the third scenario (S_3), τ_{pg} and τ_{gp} are normally distributed random

variables with mean $\mu_{\tau_{pg}} = 10\,[s]$ and $\mu_{\tau_{gp}} = 15\,[s]$ respectively; both time delays have standard deviation $\sigma = 30\,[ms]$ and change every $5\,[s]$. Fig. 3.13 and Fig. 3.14 show the plant state estimation error $\tilde{x}(t)$ and the plant state prediction error $x(t) - \check{x}(t)$ respectively.

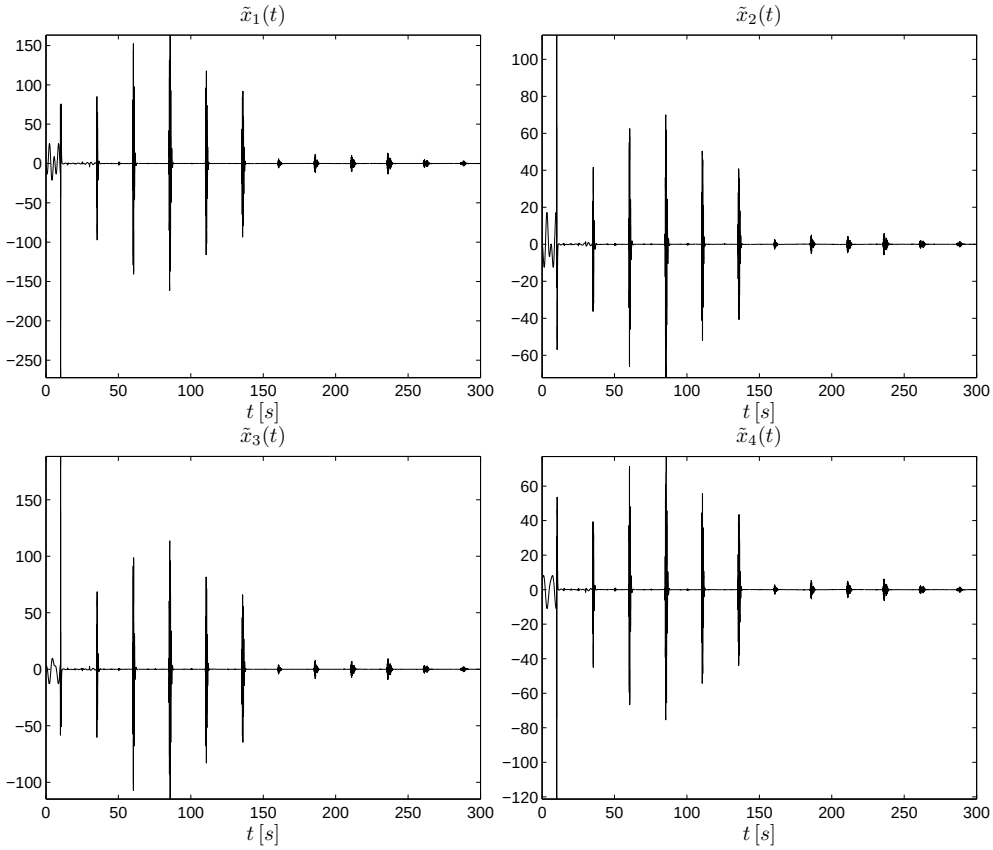

Figure 3.13: S_3: Plant state estimation error $\tilde{x}(t)$.

Some minor transient behaviors occur at times multiples of $5\,[s]$ and become smaller as time elapses. Fig. 3.15 contains the plant output $y(t)$ versus the reference signal $v(t)$ (dashed line) and the corresponding tracking error $e(t)$: the tracking error is quite small from $t = 170\,[s]$ onwards. Finally, Fig. 3.16 shows the control signal $u(t)$: similarly to Fig. 3.15, the control signal presents oscillations that become small from $t = 170\,[s]$ onwards. In spite

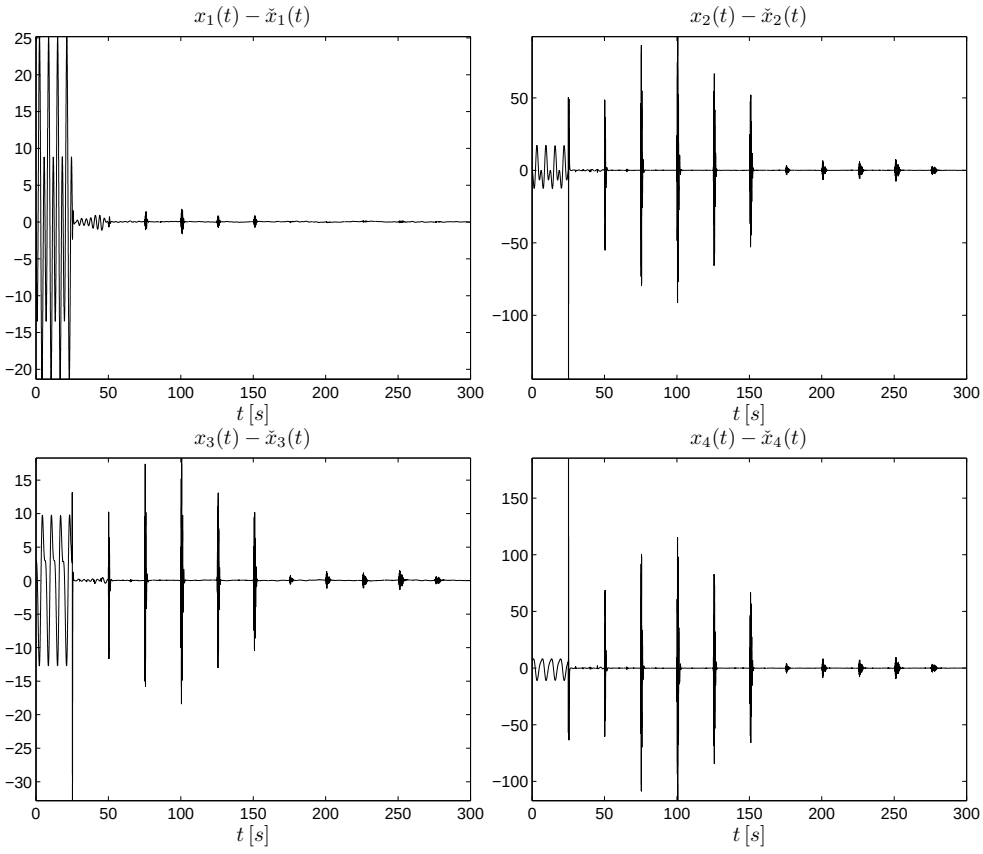

Figure 3.14: S_3: Plant state prediction error $x(t) - \check{x}(t)$.

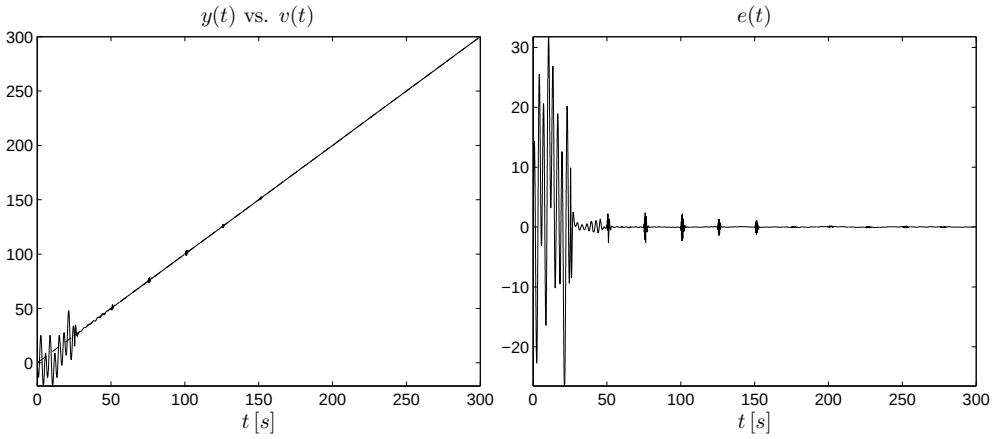

Figure 3.15: S_3: Plant output $y(t)$ versus the reference signal $v(t)$ (dashed line) and the corresponding tracking error $e(t)$.

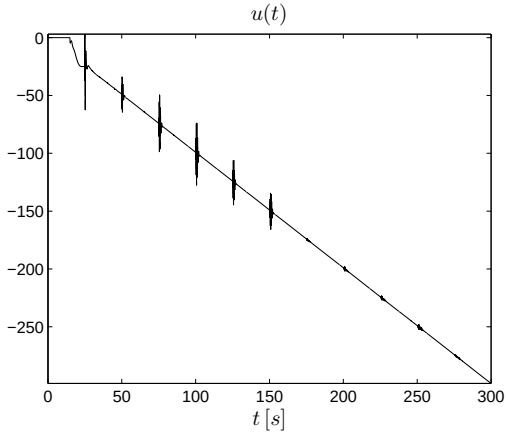

Figure 3.16: S_3: Control signal $u(t)$.

of the presence of minor transient behaviors, the delay compensation technique still works in the third scenario, allowing the plant output to track the reference signal with satisfactory accuracy.

3.4 Conclusions

An observer-based delay compensation technique has been presented and analytically examined. Its effectiveness has been tested through computer simulations in three different scenarios, involving both constant time delays (first scenario) and slightly variable time delays (second and third scenario). The proposed method is designed for constant and known time delays, thus it has been appropriately modified to also work in presence of slightly variable delays, approximating the actual time delays with their mean values. In particular, in the second and third scenario time delays are normally distributed random variables, with known mean and standard deviation.

Future research should be directed to the development of delay compensation techniques taking expressly into account the effects of variable time delays, allowing to control a plant even in presence of time delays characterized by large standard deviations. It would be also interesting to apply such control methodologies to NCSs whose systems interact each other, so that the overall system behaves in the desired manner.

Bibliography

[1] T. B. Achacoso and W. S. Yamamoto. *Ay's neuroanatomy of C. elegans for computation.* CRC Press, Boca Raton, FL, 1992.

[2] L. A. Adamic. *Lecture notes in computer science*, volume 1696, chapter The small world web, pages 443–454. Springer, New York, 1999.

[3] L. A. Adamic, R. M. Lukose, and B. A. Huberman. *Local search in unstructured networks.* Handbook of graphs and networks. Wiley-VCH, Berlin, 2003.

[4] L. A. Adamic, R. M. Lukose, A. R. Puniyani, and B. A. Huberman. Search in power-law networks. *Phys. Rev. E*, 64(046135), 2001.

[5] W. Aiello, F. Chung, and L. Lu. A random graph model for massive graphs. In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, pages 171–180, New York, 2000. Association of Computing Machinery.

[6] W. Aiello, F. Chung, and L. Lu. *Handbook of massive data sets*, chapter Random evolution in massive graphs, pages 97–122. Kluwer Academic Publishers, Dordrecht, 2002.

[7] P. F. Al-Hokayem. Stability analysis of networked control systems. Master's thesis, The University of New Mexico, ECE Dept., 2003.

[8] R. Alberich, J. Miro-Julia, and F. Rossello. Marvel universe looks almost like a real social network. *arXiv:cond-mat/0202174*, 2002.

[9] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74:47–97, 2002.

[10] R. Albert, H. Jeong, and A.-L. Barabási. Diameter of the World-Wide Web. *Nature*, 401:130–131, 1999.

[11] R. Albert, H. Jeong, and A.-L. Barabási. Error and attack tolerance of complex networks. *Nature*, 406:378–382, 2000.

[12] L. A. N. Amaral, A. Barrat, A.-L. Barabási, G. Caldarelli, P. De Los Rios, A. Erzan, B. Kahng, R. Mantegna, J. F. F. Mendes, R. Pastor-Satorras, and A. Vespignani. Virtual round table on ten leading questions for network research. *The European Physical Journal B*, 38:143–145, 2004.

[13] L. A. N. Amaral and J. M. Ottino. Complex networks. *The European Physical Journal B*, 38:147–162, 2004.

[14] L. A. N. Amaral, A. Scala, M. Barthélémy, and H. E. Stanley. Classes of small-world networks. In *Proc. Natl. Acad. Sci. USA*, volume 97, pages 11149–11152, 2000.

[15] R. M. Anderson and R. M. May. Susceptibleinfectious-recovered epidemic models with dynamic partnerships. *Journal of Mathematical Biology*, 33:661–675, 1995.

[16] C. Argento. Identification of complex networks by the method of stages. To appear in *Proc. American Control Conference*, July 2007.

[17] C. Argento and A. Tornambè. Tracking control with time delay compensation for linear time invariant networked control systems. Submitted to *Conference on Decision and Control*, December 2007.

[18] W. B. Arthur. Complexity and the economy. *Science*, 284:107–109, 1999.

[19] J. R. Banavar, A. Maritan, and A. Rinaldo. Size and form in efficient transportation networks. *Nature*, 399:130–132, 1999.

[20] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.

[21] A.-L. Barabási, R. Albert, and H. Jeong. Scale-free characteristics of random networks: The topology of the World Wide Web. *Physica A*, 281:69–77, 2000.

[22] O. Beldiman, G. Walsh, and L. Bushnell. Predictors for networked control systems. In *Proc. American Control Conference*, pages 2347–2351, Chicago, IL, June 2000.

[23] B. Bollobás. *Random graphs*. Cambridge University Press, Cambridge, 2nd edition, 2001.

[24] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the Web. *Computer Networks*, 33:309–320, 2000.

[25] D. S. Callaway, M. E. J. Newman, S. H. Strogatz, and D. J. Watts. Network robustness and fragility: Percolation on random graphs. *Phys. Rev. Lett.*, 85:5468–5471, 2000.

[26] Q. Chen, H. Chang, R. Govindan, S. Jamin, S. J. Shenker, and W. Willinger. The origin of power laws in Internet topologies revisited. In *Proc. IEEE INFOCOM*, pages 608–617, New York, June 2002.

[27] M. Y. Chow. Network-based control systems: A tutorial. In *IEEE/ASME Advanced Intelligent Mechatronics*, Monterey, CA, July 2005.

[28] G. Chowell, J. M. Hyman, S. Eubank, and C. Castillo-Chavez. Scaling laws for the movement of people between locations in a large city. *Physical Review E*, 68, 2003.

[29] R. Cohen, K. Erez, D. ben Avraham, and S. Havlin. Resilience of the Internet to random breakdowns. *Phys. Rev. Lett.*, 85(21):4626–4628, November 2000.

[30] R. Cohen, K. Erez, D. ben Avraham, and S. Havlin. Breakdown of the Internet under intentional attack. *Phys. Rev. Lett.*, 86(16):3682–3685, April 2001.

[31] T. F. Coleman and Y. Li. On the convergence of reflective Newton methods for large-scale nonlinear minimization subject to bounds. *Mathematical Programming*, 67:189–224, 1994.

[32] T. F. Coleman and Y. Li. An interior, trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization*, 6:418–445, 1996.

[33] P. Crucitti, V. Latora, M. Marchiori, and A. Rapisarda. Efficiency of scale-free networks: Error and attack tolerance. *Physica A*, 320:622–642, 2003.

[34] P. Crucitti, V. Latora, M. Marchiori, and A. Rapisarda. Error and attack tolerance of complex networks. *Physica A*, 340:388–394, 2004.

[35] R. de Castro and J. W. Grossman. Famous trails to Paul Erdős. *Mathematical Intelligencer*, 21:51–63, 1999.

[36] I. de S. Pool and M. Kochen. Contacts and influence. *Social Networks*, 1:1–48, 1978.

[37] D. J. de S. Price. Networks of scientific papers. *Science*, 149:510–515, 1965.

[38] D. J. de S. Price. A general theory of bibliometric and other cumulative advantage processes. *J. Amer. Soc. Inform. Sci.*, 27:292–306, 1976.

[39] L. Devroye. *Non-uniform random variate generation*. Springer-Verlag, 1986.

[40] S. N. Dorogovtsev and J. F. F. Mendes. Evolution of networks. *Advances in Physics*, 51:1079–1187, 2002.

[41] S. N. Dorogovtsev and J. F. F. Mendes. *Evolution of networks: From biological nets to the Internet and WWW*. Oxford University Press, 2003.

[42] J. A. Dunne, R. J. Williams, and N. D. Martinez. Food-web structure and network theory: The role of connectance and size. In *Proc. Natl. Acad. Sci. USA*, volume 99, pages 12917–12922, 2002.

[43] J. A. Dunne, R. J. Williams, and N. D. Martinez. Network structure and biodiversity loss in food webs: Robustness increases with connectance. *Ecology Letters*, 5:558–567, 2002.

[44] L. Egghe and R. Rousseau. *Introduction to Informetrics*. Elsevier, Amsterdam, 1990.

[45] P. Erdős and A. Rényi. On random graphs. *Publicationes Mathematicae*, 6:290–297, 1959.

[46] P. Erdős and A. Rényi. On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, 5:17–61, 1960.

[47] P. Erdős and A. Rényi. On the strength of connectedness of a random graph. *Acta Mathematica Scientia Hungary*, 12:261–267, 1961.

[48] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the Internet topology. *Computer Communications Review*, 29:251–262, 1999.

[49] D. A. Fell and A. Wagner. The small world of metabolism. *Nature Biotechnology*, 18:1121–1122, 2000.

[50] D. J. Felleman and D. C. Van Essen. Distributed hierarchical processing in primate cerebral cortex. *Cerebral Cortex*, 1:1–47, 1991.

[51] G. Gallo and S. Pallottino. Shortest path algorithms. *Ann. Oper. Res.*, 13:3–79, 1988.

[52] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. In *Proc. Natl. Acad. Sci. USA*, volume 99, pages 7821–7826, 2002.

[53] J. W. Grossman and P. D. F. Ion. On a portion of the well-known collaboration graph. *Congressus Numerantium*, 108:129–131, 1995.

[54] J. Guare. *Six degrees of separation: A play.* Vintage Books, New York, 1990.

[55] P. Holme, B. J. Kim, C. N. Yoon, and S. K. Han. Attack vulnerability of complex networks. *Phys. Rev. E*, 65(056109), 2002.

[56] B. A. Huberman. *The laws of the Web.* MIT Press, Cambridge, MA, 2001.

[57] K. Hwang and F. A. Briggs. *Computer architecture and parallel processing.* McGraw-Hill, New York, 1988.

[58] R. F. i Cancho and R. V. Solé. The small world of human language. In *Proc. R. Soc. London B*, volume 268, pages 2261–2265, 2001.

[59] R. Jain. *The art of computer systems performance analysis.* Wiley, New York, 1991.

[60] S. Janson, T. Luczak, and A. Rucinski. *Random graphs.* John Wiley, New York, 1999.

[61] H. Jeong, S. Mason, A.-L. Barabási, and Z. N. Oltvai. Lethality and centrality in protein networks. *Nature*, 411:41–42, 2001.

[62] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A.-L. Barabási. The large-scale organization of metabolic networks. *Nature*, 407:651–654, 2000.

[63] J. H. Jones and M. S. Handcock. An assessment of preferential attachment as a mechanism for human sexual network formation. In *Proc. R. Soc. London B*, volume 270, pages 1123–1128, 2003.

[64] V. Kalapala, V. Sanwalani, A. Clauset, and C. Moore. Scale invariance in road networks. *Physical Review E*, 73, 2006.

[65] F. Karinthy. Chains, in *Everything is different.* Budapest, 1929.

[66] J. Kautsky, N. K. Nichols, and P. Van Dooren. Robust pole assignment in linear state feedback. *Int. J. Control*, 41:1129–1155, May 1985.

[67] J. M. Kleinberg. Navigation in a small world. *Nature*, 406:845, 2000.

[68] J. M. Kleinberg. The small-world phenomenon: An algorithmic perspective. In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, pages 163–170, New York, 2000. Association for Computing Machinery.

[69] C. Koch and G. Laurent. Complexity and the nervous system. *Science*, 284:96–98, 1999.

[70] M. Kochen, editor. *The small world.* Ablex, Norwood, NJ, 1989.

[71] M. Kretschmar and M. Morris. Measures of concurrency infectious in networks and the spread of infectious disease. *Mathematical Biosciences*, 133:165–195, 1996.

[72] V. Latora and M. Marchiori. Efficient behavior of small-world networks. *Physical Review Letters*, 87, 2001.

[73] V. Latora and M. Marchiori. Is the Boston subway a small-world network? *Physica A*, 314:109–113, 2002.

[74] V. Latora and M. Marchiori. Economic small-world behavior in weighted networks. *The European Physical Journal B*, 32:249–263, 2003.

[75] S. Lawrence and C. L. Giles. Searching the World Wide Web. *Science*, 280:98–100, 1998.

[76] S. Lawrence and C. L. Giles. Accessibility of information on the Web. *Nature*, 400:107–109, 1999.

[77] F.-L. Lian, J. R. Moyne, and D. M. Tilbury. Performance evaluation of control networks. *IEEE Control Systems Magazine*, 21(1):66–83, February 2001.

[78] F. Liljeros, C. R. Edling, L. A. N. Amaral, H. E. Stanley, and Y. Åberg. The web of human sexual contacts. *Nature*, 411:907–908, 2001.

[79] I. Lopez, J. L. Piovesan, C. T. Abdallah, D. Lee, O. Martinez, M. Spong, and R. Sandoval. Practical issues in networked control systems. In *Proc. American Control Conference*, pages 4201–4206, Minneapolis, MN, June 2006.

[80] R. Luck and A. Ray. An observer-based compensator for distributed delays. *Automatica*, 26(5):903–908, September 1990.

[81] N. D. Martinez. Artifacts or attributes? Effects of resolution on the Little Rock Lake food web. *Ecological Monographs*, 61:367–392, 1991.

[82] S. Milgram. The small world problem. *Psychology Today*, 2:60–67, 1967.

[83] M. Molloy and B. Reed. A critical point for random graphs with a given degree sequence. *Random Structures and Algorithms*, 6:161–179, 1995.

[84] J. M. Montoya and R. V. Solé. Small world patterns in food webs. *J. Theor. Bio.*, 214:405–412, 2002.

[85] J. L. Moreno. *Who shall survive?* Beacon House, Beacon, NY, 1934.

[86] M. L. Van Name and B. Catchings. Benchmark testing: Advancing the art. *PC Magazine*, 1421:13, 1996.

[87] M. E. J. Newman. Scientific collaboration networks: I. Network construction and fundamental results. *Physical Review E*, 64(016131), 2001.

[88] M. E. J. Newman. Scientific collaboration networks: II. Shortest paths, weighted networks, and centrality. *Physical Review E*, 64(016132), 2001.

[89] M. E. J. Newman. The structure of scientific collaboration networks. In *Proc. Natl. Acad. Sci. USA*, volume 98, pages 404–409, 2001.

[90] M. E. J. Newman. Ego-centered networks and the ripple effect. *Social Networks*, 25:83–95, 2003.

[91] M. E. J. Newman. *Handbook of graphs and networks*, chapter Random graphs as models of networks. Wiley-VCH, 2003.

[92] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45:167–256, 2003.

[93] M. E. J. Newman. *The new Palgrave encyclopedia of economics*, chapter The mathematics of networks. Palgrave Macmillan, Basingstoke, 2nd edition, in press.

[94] M. E. J. Newman, S. Forrest, and J. Balthrop. Email networks and the spread of computer viruses. *Phys. Rev. E*, 66(035101), 2002.

[95] M. E. J. Newman, S. H. Strogatz, and D. J. Watts. Random graphs with arbitrary degree distributions and their applications. *Physical Review E*, 64(026118), 2001.

[96] A. W. Olbrot. Observability and observers for a class of linear systems with delays. *IEEE Trans. Aut. Contr.*, AC–26(2):513–517, April 1981.

[97] D. N. Pandya and E. H. Yeterian. Architecture and connections of cortical association areas. *Cerebral Cortex*, 4:3–61, 1985.

[98] R. Pastor-Satorras, A. Vázquez, and A. Vespignani. Dynamical and correlation properties of the Internet. *Phys. Rev. Lett.*, 87(258701), 2001.

[99] A. G. Phadke and J. S. Thorp. *Computer relaying for power systems*. Wiley, New York, 1988.

[100] N. J. Ploplys, P. A. Kawka, and A. G. Alleyne. Closed-loop control over wireless networks. *IEEE Control Systems Magazine*, 24(3):58–71, June 2004.

[101] J. J. Potterat, L. Phillips-Plummer, S. Q. Muth, R. B. Rothenberg, D. E. Woodhouse, T. S. Maldonado-Long, H. P. Zimmerman, and J. B. Muth. Risk network structure in the early epidemic phase of HIV transmission in Colorado Springs. *Sexually Transmitted Infections*, 78:i159–i163, 2002.

[102] A. Rapoport. Contribution to the theory of random and biased nets. *Bulletin of Mathematical Biophysics*, 19:257–277, 1957.

[103] S. Redner. How popular is your paper? An empirical study of the citation distribution. *The European Physical Journal B*, 4:131–134, 1998.

[104] L. Sattenspiel and C. P. Simon. The spread and persistence of infectious diseases in structured populations. *Mathematical Biosciences*, 90:341–366, 1988.

[105] J. W. Scannell. Determining cortical landscapes. *Nature*, 386:452, 1997.

[106] J. W. Scannell, C. Blakemore, and M. P. Young. Analysis of connectivity in the cat cerebral cortex. *The Journal of Neuroscience*, 15:1463–1483, February 1995.

[107] J. Scott. *Social network analysis: A handbook*. Sage Publications, London, 2nd edition, 2000.

[108] P. O. Seglen. The skewness of science. *J. Amer. Soc. Inform. Sci.*, 43:628–638, 1992.

[109] P. Sen, S. Dasgupta, A. Chatterjee, P. A. Sreeram, G. Mukherjee, and S. S. Manna. Small-world properties of the Indian railway network. *Physical Review E*, 67, 2003.

[110] E. Sivan, H. Parnas, and D. Dolev. Fault tolerance in the cardiac ganglion of the lobster. *Biological Cybernetics*, 81:11–23, 1999.

[111] T. Skeie, S. Johannessen, and C. Brunner. Ethernet in substation automation. *IEEE Control Systems Magazine*, 22(3):43–51, June 2002.

[112] J. E. Smith. Characterizing computer performance with a single number. *Commun. ACM*, 31(10):1202–1206, 1988.

[113] R. Solomonoff and A. Rapoport. Connectivity of random nets. *Bulletin of Mathematical Biophysics*, 13:107–117, 1951.

[114] O. Sporns, G. Tononi, and G. M. Edelman. Theoretical neuroanatomy: Relating anatomical and functional connectivity in graphs and cortical connection matrices. *Cerebral Cortex*, 10:127–141, 2000.

[115] S. H. Strogatz. Exploring complex networks. *Nature*, 410:268–276, 2001.

[116] K. Thulasiraman and M. N. S. Swamy. *Graphs: Theory and algorithms.* John Wiley & Sons, 1992.

[117] Y. Tipsuwan and M. Y. Chow. Control methodologies in networked control systems. *Control Engineering Practice*, 11(10):1099–1111, 2003.

[118] A. Tornambè. *Discrete-event system theory: An introduction.* World Scientific, 1995.

[119] J. Travers and S. Milgram. An experimental study of the small world problem. *Sociometry*, 32:425–443, 1969.

[120] A. Vazquez. Statistics of citation networks. *arXiv:cond-mat/0105031*, 2001.

[121] A. Vazquez, R. Pastor-Satorras, and A. Vespignani. Large-scale topological and dynamical properties of the Internet. *Phys. Rev. E*, 65(066130), 2002.

[122] G. Walsh, H. Ye, and L. Bushnell. Stability analysis of networked control systems. *IEEE Trans. Contr. Syst. Technol.*, 10(3):438–446, May 2002.

[123] S. Wasserman and K. Faust. *Social network analysis: Methods and applications*. Cambridge University Press, Cambridge, 1994.

[124] D. J. Watts. *Small worlds*. Princeton University Press, Princeton, 1999.

[125] D. J. Watts, P. S. Dodds, and M. E. J. Newman. Identity and search in social networks. *Science*, 296:1302–1305, 2002.

[126] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.

[127] J. G. White, E. Southgate, J. N. Thomson, and S. Brenner. The structure of the nervous system of the nematode Caenorhabditis elegans. *Phil. Trans. R. Soc. B*, 314:1–340, 1986.

[128] H. Ye and G. Walsh. Real-time mixed-traffic wireless networks. *IEEE Trans. Ind. Electron.*, 48(5):883–890, October 2001.

[129] H. Ye, G. Walsh, and L. Bushnell. Wireless local area networks in the manufacturing industry. In *Proc. American Control Conference*, pages 2363–2367, Chicago, IL, June 2000.

[130] M. P. Young. The organization of neural systems in the primate cerebral cortex. *Phil. Trans. R. Soc. B*, 252:13–18, 1993.

[131] W. Zhang, M. S. Branicky, and S. Phillips. Stability of networked control systems. *IEEE Control Systems Magazine*, 21(1):84–99, February 2001.