HELSINKI UNIVERSITY OF TECHNOLOGY
Department of Computer Science and Engineering
Laboratory for Theoretical Computer Science

Satu Virtanen

Properties of nonuniform random graph models

Licentiate's thesis submitted in partial fulfillment of the requirements for the degree of Licentiate of Science in Technology

Supervisor Professor Pekka Orponen

Espoo, March 31, 2003

TEKNILLINEN KORKEAKOULU | LISENSIAATTITYÖN TIIVISTELMÄ

Tekijä:	Satu Virtanen		
Työn nimi:	Epäuniformien satunnaisverkkomallien ominaisuuksia		
English title:	Properties of nonuniform random graph models		
Päivämäärä:	31 maaliskuuta, 2003	**Sivumäärä:**	xii + 121
Osasto:	Tietotekniikan osasto		
Professuuri:	T-79 Tietojenkäsittelyteoria		
Työn valvoja:	Professori Pekka Orponen		

Tämä työ tarkastelee luonnollisen kaltaisten verkkojen tuottamiseen tarkoitettuja satunnaisverkkomalleja ja niiden ominaisuuksia. Työ on kartoitustutkimus alan kirjallisuudesta ja siihen sisältyy kokeellinen osa. Työ tarjoaa laajan katsauksen yleisimpiin malliperheisiin, mukaanlukien perinteiset uniformit satunnaisverkkomallit. Myös joitakin deterministisiä malleja käsitellään.

Viime vuosikymmenen loppupuolella esitetyistä, paljon tutkimusta herättäneistä lähestymistavoista käsitellään nk. pieni maailma -verkkoja, joilla pyritään saavuttamaan sosiaalisten verkkojen ominaisuuksia, sekä astelukujakaumaltaan skaalattomia verkkoja, joilla pyritään mallintamaan mm. tietoliikenneverkkoja.

Työn alkupuolella paneudutaan luonnollisten verkkojen mallinnukseen esimerkkitapausten kautta. Työssä käsitellään laajasti Internet-verkolle ja WWW:lle ehdotettuja malleja sekä näistä saatuja mittaustuloksia. Matemaattisten mallien käsittely alkaa johdannolla verkkoteoriaan, jonka käsitteistöä käyttäen tarkastellaan erilaisten malliperheiden matemaattisia kuvauksia ja analyyttisia ominaisuuksia.

Työn loppuosa koostuu kokeellisista tuloksista. Kokeissa tutkitaan erilaisten satunnaisverkkomallien ominaisuuksia toteutetun ohjelmiston avulla. Matemaattisten mittareiden lisäksi kokeissa tutkitaan verkkomallin vaikutusta verkon suurimman klikin koon määräävän algoritmin suoritusaikaan.

Verkkomallien ohella työssä käsitellään verkkojen klusterointiin tarkoitettuja algoritmeja, joiden avulla pystytään tunnistamaan semanttisesti merkittäviä osakokonaisuuksia annetusta mielivaltaisesta verkosta. Koska perinteiset klusterointialgoritmit, jotka pyrkivät osittamaan koko verkon, eivät skaalaudu hyvin suurille verkoille kuten WWW-verkkomallille, paikalliseen hakuun perustuvat algorimit ovat tarpeellisia. Työssä esitetään heuristinen menetelmä suurten, mahdollisesti osin tuntemattomien verkkojen paikalliseen klusterointiin.

Avainsanat: satunnaisverkko, verkkoalgoritmi, verkkoklusterointi, verkkomallinnus

HELSINKI UNIVERSITY OF TECHNOLOGY ABSTRACT OF LICENTIATE'S THESIS

Author:	Satu Virtanen
Title of thesis:	Properties of nonuniform random graph models
Finnish title:	Epäuniformien satunnaisverkkomallien ominaisuuksia
Date:	March 31, 2003 **Pages:** xii + 121
Department:	Department of Computer Science and Engineering
Chair:	T-79 Theoretical Computer Science
Supervisor:	Professor Pekka Orponen

This thesis is a survey on network models designed to produce graphs that resemble natural networks. Unlike artificially generated networks, natural networks are graphs that have been constructed based on some phenomenon or object of the real world. The thesis includes two extensive case studies of natural networks that emerge from engineering applications: the network model of the router-level Internet and the Web graph, which is a model of the World Wide Web.

Several different models for generating such networks are discussed. After a brief summary of basic graph theory, the traditional model of uniform random graphs is presented with generalizations. Two recent models of natural graphs are discussed in detail: the small-world networks that capture characteristics of social networks, and the scale-free networks that imitate real-world communication networks. Several variations of both models are presented, including some deterministic models.

After studying the mathematical descriptions of the models and some analytically derived properties, experimental work is documented. Properties of different network models are examined by software implementations of the model descriptions. In addition to calculating some graph theoretical metrics, the algorithmic implications of the network models are studied through the running times of an algorithm that determines the order of a maximum clique in a given graph.

The thesis also contains a brief review on clustering algorithms for graphs. The goal of clustering is to identify semantically meaningful entities from arbitrary graphs. The traditional approach to clustering is to split the given graph into clusters starting from the entire graph. This does not scale well to very large graphs, and therefore algorithms that employ local search are of interest. In this thesis, heuristic methods for finding clusters from large and possibly unknown graphs are proposed.

Keywords: graph algorithm, graph clustering, network modeling, random graph

Preface

I thank my supervisor, professor Pekka Orponen, for encouraging me to work on such a diverse and cross-disciplinary topic as network modeling, as well as for his endless support, surprising enthusiasm, and genuine interest in my work. His insight and guidance have been essential ingredients of this work. I am also grateful to Kari Eloranta, the reviewer of this thesis, for valuable comments and feedback during the revision process.

In addition, I thank Esko Nuutila for his support in many details of implementation and measurements, and Kosti Rytkönen for his help with graph visualization as well as many fruitful conversations. I also thank the Laboratory of Theoretical Computer Science, especially the head of the laboratory, professor Ilkka Niemelä, for providing me with a supporting and functional working environment. For proof-reading and helpful comments, I thank my friends and fellow students Petteri Kaski and Timo Latvala.

This document has been typeset on `emacs` using LaTeX, most figures drawn with `xfig` and embedded in the document with `transfig`. The experiments are documented with the help of `gnuplot`. In order to use these tools efficiently, I have received help from several coworkers and am grateful for that assistance.

This research was supported in part by the Academy of Finland under grant 81120 and Helsinki Graduate School in Computer Science and Engineering (HeCSE).

Espoo, March 31, 2003

Satu Virtanen

Contents

Abbreviations and notations

AS	Autonomous Systems (the Internet domains)
BA	Barabási-Albert graph generation model
CBA	Barabási-Albert model with tunable clustering
ER	Erdős-Rényi model of random graphs
IMDb	Internet Movie Database
SCC	Strongly connected component
SWS	Solvable Watts-Strogatz graph generation model
WS	Watts-Strogatz graph generation model
WWW	World Wide Web

G	graph $G = (V, E)$
V	set of vertices in a graph
E	set of edges in a graph
n	the number of vertices $\lvert V\rvert$ (order of a graph)
m	the number of edges $\lvert E\rvert$ (size of a graph)
(u, v)	edge connecting vertices u and v
$\langle u, v\rangle$	directed edge from vertex u to vertex v
$\Gamma(v)$	neighborhood of a vertex v
$\deg(v)$	degree of a vertex v
K_n	complete graph of n vertices
$K_{n,k}$	complete bipartite graph on $n + k$ vertices
$C_{n,k}$	$2k$-regular *circulant* graph on n vertices
P_n	a graph on n vertices forming a simple path with $n - 1$ edges

$[a, b]$	closed interval from a to b
(a, b)	open interval from a to b
$[a, b)$	half-open interval containing a but not b
$(a, b]$	half-open interval containing b but not a
$f(x) \sim g(x)$	similar functions; $\lim_{x\to\infty} f(x)/g(x) = 1$
$x \propto y$	x is proportional to y
$\mathbf{I}$	*identity* matrix
$\mathbf{J}$	*unit* matrix; all elements are equal to one
$\mathbf{A}$	*adjacency* matrix of a graph

Chapter 1

Introduction

Networks are common models of complex systems, approachable by methods of graph theory and statistical mechanics. Over recent years, the properties of natural network models have become an intensive field of study and also a recurring theme of popular science (see for example [11, 133]). The purpose of constructing models for natural networks is to study their structure in more general level than through a single instance, and to study different phenomena in relation to the networks, such as epidemic spreading or robustness of the network when imposed to random failures or deliberate attack.

The goal of this thesis is to examine the properties and practical implications of different network models in the form of a comprehensive survey of recent work. This includes the traditional models of uniform random graphs [43, 44, 54] and the recently proposed *small-world* networks of Watts and Strogatz [134] and the *scale-free* network model of Barabási and Albert [12]. We discuss variations proposed of these models along with alternative approaches, first with emphasis on the construction and later, in the experimental part of the work, studying the structural properties achieved with the constructions.

The algorithmic implications of network structure are of special interest, as in practice the performance of an algorithm is only relevant for all practical inputs instead of all imaginable inputs. For example, a routing algorithm should work well on the communication networks for which it is intended, and hence if a characterization of such networks is available, the routing should be optimized with respect to that characterization instead of optimizing it for any mathematically possible network structure. We review some results obtained on the behavior of for example *shortest-path* and *graph coloring* algorithms, and augment the existing experimental studies by examining the running times of a *maximum clique* algorithm for different network models.

Another interesting problem is *data clustering* for graphs. Several algorithms exist to cluster a graph based on adjacency information that produce either one clustering for the entire graph or a hierarchy of possible clusterings. These methods are only feasible for small graphs and we do not expect them to scale well for very large graphs. In some application areas such as locating clusters from the World Wide Web, finding clusters for the entire network is neither feasible nor relevant, and hence local methods are required. We propose a heuristic for local search that identifies clusters from massive and possibly partially unknown graphs.

To examine the graph models and their properties experimentally, we have implemented a framework under which graphs may be generated stochastically, including several of the models discussed in the preceding parts of the survey. New models can easily be added to the toolset and variations of the existing models are simple to include. We have also included functionality to derive different measures for graphs either generated with the toolset or given as properly formatted input to the toolset. We have analyzed the generation models with respect to some of the measures proposed in recent literature, such as the *characteristic path length*, *clustering coefficient*, and the *degree distribution* of the graphs.

The text is organized as follows. In Chapter 2, network modeling is introduced through examples, and the fundamental graph theoretical terminology is summarized. The network modeling efforts on the Internet and the World Wide Web are reviewed as case studies. Chapter 3 summarizes mathematical models of networks, starting from the traditional random networks of Erdős and Rényi [43, 44], and proceeding to the various models proposed to capture essential properties of natural networks.

Chapter 4 addresses in general properties of these families of random networks, such as spectral properties, error tolerance, and behavior of random walks. Some important graph problems and algorithms are also discussed. In particular, Graph Coloring and algorithms involving shortest paths are reviewed. Clustering of graphs is addressed with more detail, also presenting new heuristics for finding clusters locally. The conducted experiments, mainly studying the properties of the generation models, are described and their results are presented in Chapter 5. Closing remarks, including possibilities for further work are discussed in Chapter 6.

Chapter 2

Network modeling

A network consists of a set of *nodes* and usually several *connections* between the nodes. The nodes or the connections may contain some additional information, such as labels or weights. Networks are common models of complex systems in many fields of science; there are numerous examples in engineering, biology, sociology, and even linguistics, some of which will be presented in Section 2.1 to provide a concrete view on network modeling.

Representing a complex system such as a human cell or the epidemic spreading of a virus as a collection of nodes and connections obviously cannot capture all information present in the original system. There are limitations in all models; it is the task of the modeler to carefully select the features of the modeled phenomenon that are relevant with respect to the goal. The simpler the model, the more fluent the calculations. However, excess simplicity tends to limit the usefulness of the model.

2.1 Natural networks

We begin our survey of network modeling by presenting examples of natural networks that have been intensively studied and will be encountered several times in this text. Both analytical and empirical results on these network models are abundant. As an example of a biological network model, we present the neural network of a widely studied worm, the *Caenorhabditis elegans*. It is a tiny nematode, only about one millimeter in length, having only 959 cells (see [133] and the references therein). It has been a target of intensive research for decades now; the Nobel Prize in Physiology or Medicine 2002 was awarded to three researchers who have all been working on this nematode. The entire genome of the *C. elegans* has been among the first to be mapped and almost all connections between its nerve cells

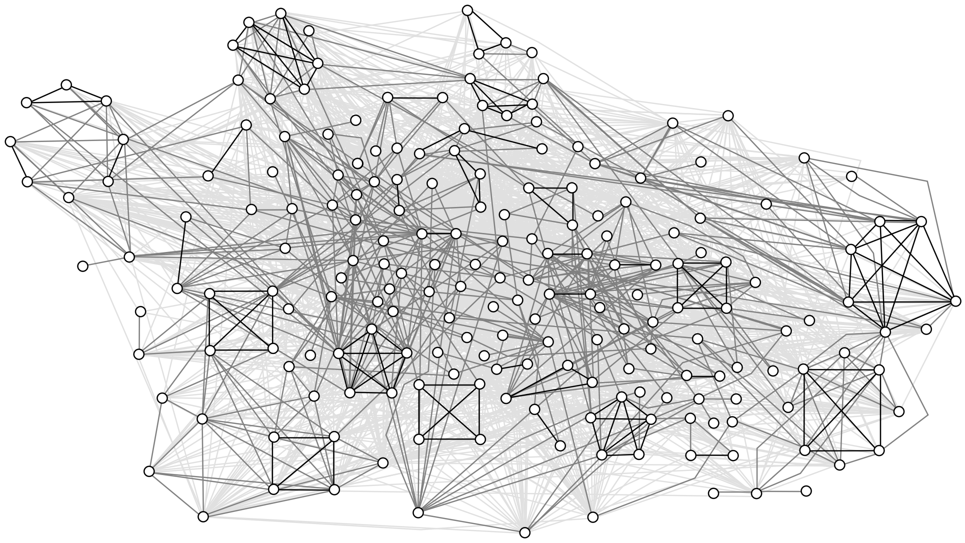


FIGURE 2.1: The neural network of the *C. elegans* modeled as a simple network of cells (circles) and synaptic connections (lines), ignoring all qualitative information of the biological system under study.

are known. It even has a dedicated website at `http://elegans.swmed.edu/`.

Following the example of Duncan Watts [133], we obtained data of the synaptic connections available at the website and constructed a network with all 202 neurons as equal nodes and and the 1,954 neuron synapses and gap junctions are modeled as equal connections. If more than one synaptic connection or gap junction exists between two neurons, only one connection is present in the model for simplicity. The average number of connections we obtained is approximately 19, and the longest separation between vertices (when the minimal possible number of connections is traversed) is five connections. Figure 2.1 shows the network we obtained from the data available online. The lines represent the connections and the dots are the nodes. Apparently, the model is not specifically designed for heavy biological analysis, as most of the biological details have been stripped. The object of study is the structure of the resulting network. Other common biological network models are cell models, representing chemical interactions within a cell, for instance in DNA replication. Among others, Arita [9] believes that reconstruction of metabolism will be a major research area in biology and proposes a network model to study metabolic structure; the topology of metabolic networks has been studied by Jeong et al. [69].

An engineering example is the electrical power grid of the western areas of the United States, from the West Coast to the Rocky Mountains. This network model was studied by Duncan Watts [133] as an example of a network model in engineering in his doctoral study, as the data was readily available in appropriate format. The 4,941 nodes of the network are generators, transformers and other stations related to the production and transmission of electrical power. The connections are the electric power lines, ignoring lines connecting the transmission network to the consumers. There are on average 2.67 connections per each node in the network.

Although in reality the machinery and functionality of different types of nodes varies a great deal, Watts chose not to classify or label the nodes accordingly, but simplified the network by treating all nodes as equal. Although different transmission lines certainly have different length, direction, and capacity, all connections were also treated as equal: unweighted and undirected. This obviously delimits the usefulness of the model. For instance, the effects of the failure of one power line cannot be simulated with the model, as the information on which way the line works has been left out of the model. However, some meaningful topological questions may be addressed even when the dynamical nature and properties of the nodes and connections have been reduced. This has been a common practice in much of the recent work on natural networks, and reflects on the approach taken in Chapter 3.

Other engineering networks are, for instance, the connections between international airports worldwide, or the public transportation network of any metropolis. In sociology, network models that represent social connections within a population have been studied already in the 1940s (see [107] and the references therein). As it is practically impossible to measure the number of social contacts a person has in any but subjective manner, exact network models of social interactions are rare. Popular examples of such are collaboration networks, manageable by documentation. For instance, citations in scientific publications form a network. The nodes are the authors, and a connection appears as one author cites a paper by another. Yet another network emerges if a connection is placed between two authors as they first publish together. A famous example are the *Erdős numbers*: an author who has published something with the late Pál Erdős, a famous Hungarian mathematician, has Erdős number one, an author who has not published with Erdős himself, but with someone of Erdős number one, has Erdős number two, and so forth.[1] Some recent research on scientific collaboration networks is summarized in Section 5.3 in conjunction with our own observations on a coauthorship network.

Another well-documented social network is the collaboration network of movie stars, based on the Internet Movie Database (IMDb), in which more than one million filmographies of cast and crew members have been stored. The database is accessible online at `http://www.imdb.com` and covers most of the movies made

[1] In 1999, there were 492 authors with Erdős number one (according to [133]), but as Erdős has continued publishing *post mortem*, this number is still growing. The website of the Erdős number project is at `http://www.oakland.edu/~grossman/erdoshp.html`.

since 1891. The information is not limited to big Hollywood productions; several foreign movies, productions of independent studios, television productions, and even future releases are covered by the database. The network that results in taking all the actors as nodes and placing a connection between each pair of actors who have appeared together in at least one movie, is an example of a large collaboration network. The biggest *connected component* of this network, that is, the part in which there exists a "chain of collaboration" between any two actors[2] captured 250,000 Hollywood actors appearing in about 110,000 movie titles in 1997, when Duncan Watts first studied the network [133]. The average number of connections per actor was approximately 61.

Motter et al. [98] have studied networks of linguistics and cognitive science: the nodes are the words of a certain language and a connection exists between two words if they are synonyms. The example network of Motter et al. was based on a thesaurus of the English language.[3] The network consists of $30,000$ entries which are on the average linked to 60 of the other entries. Other widely studied and obvious network models from the field of computer science are the Internet and the World Wide Web, discussed in more detail as case studies in Sections 2.3 and 2.4 respectively.

2.2 Graph theory

In order to discuss networks formally, the notion of a *graph* is necessary. Only the most basic graph theoretical concepts are introduced here. The notations and naming conventions are by no means unambiguous; numerous different notations are used in the literature. See for example Reinhard Diestel's book "*Graph Theory*" [37] for a comprehensive introduction.

A *graph* $G = (V, E)$ consists of two distinct sets: $V = \{v_1, \dots, v_n\}$ contains the *vertices* of the graph, and E contains the *edges* of the graph. Each edge is a pair of vertices. The *order* $|G|$ of the graph is the number of vertices, denoted by $|V|$ or briefly n. The number of edges is called the *size* of the graph, denoted by $|E|$ or m. The vertices may be *labeled*, often by the integers $1, \dots, n$. A numerical value can be attached to each vertex (called a *fitness*, often $f : V \to \mathbb{R}$) or edge (called a *cost* or a *capacity*, often $c : E \to \mathbb{R}$). *Directed* edges are ordered pairs $\langle u, v \rangle$, where $u \in V$ is the *source* of the edge and $v \in V$ the *target*, in which case the graph is a *directed graph*. Unless explicitly stated, in this text edges are unordered pairs $\{u, v\}$; such edges are called *undirected* and are denoted by (u, v).

Graphs without duplicate edges are *simple*, and those where several edges may connect two vertices u and v are *multigraphs*. The graphs considered in this text,

[2] The notion of a connected component will be formalized in Section 2.2.
[3] Available online at `ftp://ibiblio.org/pub/docs/books/gutenberg/etext02/mthes10.zip`.

unless explicitly stated otherwise, contain no duplicate or reflexive[4] edges. Therefore only one edge may connect each *distinct* pair of vertices. Hence the maximum number of edges present is $\binom{n}{2} = \frac{1}{2}n(n-1)$. A graph that contains all these $\binom{n}{2}$ edges is called the *complete* graph, denoted by K_n. The *density* of a graph $G = (V, E)$ is defined as the ratio of edges present in comparison to K_n, that is $\delta = \frac{m}{\binom{n}{2}} = \frac{2m}{n(n-1)}$.

In a *bipartite graph* $G = (U \cup V, E)$, the sets U and V are nonempty and disjoint ($U \cap V = \emptyset$) and edges may only appear between these sets: $E \subseteq \{(u, v) \mid u \in U, v \in V\}$. Hence there are no edges connecting vertices on either "side" of the graph to vertices on the same side. This generalizes of course to finer partitions of the vertex set than simply $U \cup V$, leading to the definition of a *k-partite* graph. In the *complete bipartite graph* $K_{n,k}$ all possible edges connecting U, $|U| = n$, to V, $|V| = k$, are present in the graph: $E = U \times V$.

Two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are *isomorphic*, if there exists a bijective mapping $f : V_1 \to V_2$ (called an *isomorphism*) such that $(v, w) \in E_1$ if and only if $(f(v), f(w)) \in E_2$. We write $G_1 \cong G_2$ to indicate that an isomorphism exists for G_1 and G_2. A *subgraph* $H = (V', E')$ of $G = (V, E)$ is a graph for which $V' \subseteq V$ and $E' \subseteq E$ with the additional restriction that if $(u, v) \in E'$, then $u \in V'$ and $v \in V'$. We write $H \subseteq G$ when H is a subgraph of G. Any subset $V' \subseteq V$ of vertices induces a subgraph $H = (V', E')$ such that $E' = \{(u, v) \mid u, v \in V', (u, v) \in E\}$. Such a subgraph is called the *induced subgraph* of V' in G. Note that an induced subgraph necessarily contains *all* edges in G that have both endpoints in V', whereas a general subgraph may exclude some or all of these edges.

A *clique* is a subgraph H induced by $V' \subseteq V$, $|V'| = h$, such that $H \cong K_h$. An *independent set* is the vertex set V' of an induced subgraph $H = (V', E')$ such that $E' = \emptyset$. A clique in a graph $G = (V, E)$ is an independent set of the complement of the graph, $\overline{G} = (V, \overline{E})$, where $\overline{E} = \{(u, v) \mid (u, v) \notin E, u \neq v\}$. Determining whether a clique or an independent set of given order h exist in a given graph are ***NP**-complete problems.*[5]

The *neighborhood* of a vertex $v \in V$, denoted by $\Gamma(v) \subseteq V$, is the set of vertices $\Gamma(v) = \{u \mid (v, u) \in E\}$. Note that a vertex itself is not considered to be a part of its neighborhood. If $u \in \Gamma(v)$, u and v are said to be *adjacent*. A graph is easily represented by its *adjacency matrix* $\mathbf{A}$: for $V = \{v_1, v_2, \ldots, v_n\}$, element a_{ij} of $\mathbf{A}$ is one if $(v_i, v_j) \in E$ and zero otherwise:

$$a_{ij} = \begin{cases} 1 \text{ if } (v_i, v_j) \in E \\ 0 \text{ if } (v_i, v_j) \notin E \end{cases} \tag{2.1}$$

[4]A *reflexive* edge (v, v) connects a vertex v to itself. Such edges are sometimes called *loops*.

[5]See [52] or [113] for more on **NP**-completeness.

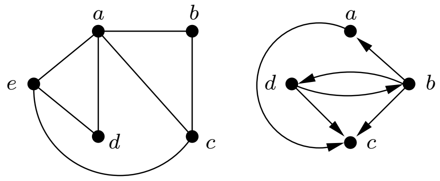


FIGURE 2.2: Examples of graphs: an undirected graph $G_1 = (V_1, E_1)$ with $V_1 = \{a, b, c, d, e\}$ and $E_1 = \{(a, b), (a, c), (a, d), (a, e), (b, c), (c, e), (d, e)\}$, and a directed graph $G_2 = (V_2, E_2)$ with $V_2 = \{a, b, c, d\}$ and $E_2 = \{\langle a, c\rangle, \langle b, a\rangle, \langle b, c\rangle, \langle b, d\rangle, \langle d, b\rangle, \langle d, c\rangle\}$.

For an undirected graph, $\mathbf{A}$ is always symmetric. If edges are assigned capacities, these can be used to form a similar matrix with the weight of the edge (i, j) as a_{ij}, placing zeroes where edges are not present. Graphs are often depicted by drawing vertices as dots and edges as lines connecting dots according to the adjacency relation. In a directed graph, the edges are drawn as arrows. See Figure 2.2 for examples.

The *degree* of a vertex is the number of edges connected to it, i.e., the size of its neighborhood: $\deg(v) = |\Gamma(v)|$. For a directed graph, the *in-degree* $\deg_{\text{in}}(v)$ of a vertex v is the number of incoming edges (u, v) and the *out-degree* $\deg_{\text{out}}(v)$ respectively the number of outgoing edges (v, w). If all the vertices of a graph have the same degree $\deg(v) = k$, the graph is said to be *k-regular*. The *average degree* in a graph is denoted by $\bar{k}$ and the *maximum degree* by Δ. The average degree is defined as $\overline{k} = 2m/n = \delta(n-1)$ by the definition of density. A *degree sequence* of a graph is a vector $\boldsymbol{d} \in \mathbb{Z}^n$ such that $\boldsymbol{d} = (\deg(v_1), \deg(v_2), \ldots, \deg(v_n))$. This can be made unique by sorting the elements in decreasing (or increasing) order. The *degree distribution* of a graph is a function $P(k)$ that assigns each $k \in [0, n)$ the probability that an arbitrary vertex $v \in V$ has exactly k neighbors, $\Pr[\deg(v) = k]$.

Two vertices are *connected* if there exists a *path* $P \subseteq E$ of consecutive edges in the graph between them (in a directed graph, edge orientation is obeyed). A maximal set of connected vertices in a graph G induces a subgraph that is a *component* of G. A graph is *connected* if all vertices are pairwise connected. If a graph has more than one component, it is *disconnected*. If there are at least two vertex-disjoint paths between any pair of vertices, the graph is *biconnected*. The *connected component* of a graph is the subgraph induced by the maximum[6] vertex set $S \subseteq V$ where all vertices in S are connected; similarly the *biconnected component* is the subgraph induced by the maximum vertex set $S' \subseteq V$ where all vertices are connected by at least two disjoint paths.

A *edge cut* for a graph $G = (V, E)$ is an edge set $E' \subseteq E$ such that $G = (V, E \setminus E')$ is disconnected. Similarly a *vertex cut* is a set of vertices whose removal disconnects

[6] A *maximal* set with respect to some property is such that no element can be added without breaking the property, whereas a *maximum* set is one of the largest order (not necessarily unique).

the graph. A single edge that forms an edge-cut is called a *cut-edge* or a *bridge*, and a single vertex that forms a vertex cut is called a *cut-vertex* or an *articulation point*. A *minimum edge cut* of a connected graph $G = (V, E)$ is an edge cut $C \subseteq E$ of minimum order. The corresponding optimization problem, MINIMUM CUT, is solvable in polynomial time. A graph G is *k-edge-connected* if at least k edges need to be removed to disconnect G. The maximum k such that G is k-connected is called the *edge-connectivity* of G and denoted by $\underline{k}(G)$. The *vertex-connectivity* of a graph is defined equivalently.

Path length is defined as the number of edges on the path, $|P|$. A *simple path* does not visit the same vertex twice. In this text, all paths are simple unless otherwise stated. A graph of order n that contains only those $n-1$ edges that are needed to form a simple path between the vertices is denoted by P_n. A *shortest path* between two vertices is the one with the least number of edges traversed. Note that this is not necessarily unique. The *distance* $d(u, v)$ between vertices u and v is equal to the length of the shortest path from u to v in G. If no such path exists in G, the distance is infinite (by convention): $d(u, v) = \infty$. If edges are assigned weights $w : E \to \mathbb{R}$, a weighted distance $dist(u, v) = \sum_{e \in P} w(e)$ can be used to define an alternative measure of path length.

The maximum value of $d(u, v)$ from a *fixed* vertex u to any other vertex $v \in V$ is called the *eccentricity* of u. The minimum eccentricity over *all* vertices is called the *radius* of the graph G and is denoted by $r(G)$. The maximum $d(u, v)$ for all vertex pairs is called the *diameter* of the graph, $\text{diam}(G)$. Another common path-related quantity is the *average path length* $\mathcal{L}(G)$ (also called the *characteristic path length*), which is the average value of the distance $d(u, v)$ over all possible pairs $\{u, v\}$, of which there are $\binom{n}{2}$ in total:

$$\mathcal{L}(G) = \frac{2}{n(n-1)} \sum_{\{u,v\} \subseteq V} d(u, v). \tag{2.2}$$

A *cycle* is a simple path that begins and ends at the same vertex. Similarly to path length, the length of a cycle is defined as the number of edges traversed on the cycle. The length of the shortest cycle is called the *girth* g of the graph. If reflexive and multiple edges are excluded, $g \geq 3$. A cycle of length three is called a *triangle*; in the literature, the term *triad* also appears. A graph that consists of n vertices and only those n edges that are needed to form a cycle of all the vertices is denoted by C_n or by $C_{n,1}$. If a graph does not contain a cycle of any length, it is said to be *acyclic*. The girth of an acyclic graph is by convention infinite.

An acyclic graph is a called a *forest*. A connected forest is called a *tree*. A *subtree* of a graph is a connected subset of edges that does not form a cycle. A subtree that includes all vertices is called a *spanning tree* of the graph. A spanning tree has necessarily $n-1$ edges. If edges are assigned weights, the spanning tree with smallest total weight is called the *minimum spanning tree*. Note that there may exist several minimum spanning trees that may even be edge-disjoint.

A tree is *rooted* if one vertex is chosen as the *root*. In a rooted tree, vertex w is the parent of vertex v if and only if w is the second vertex on the unique simple path from v to the root vertex. The other vertices on that path are called *ancestors* of v in the tree. Hence the root has no parent or ancestors. A vertex with no other neighbors but the parent is called a *leaf*.

The *spectrum* of a graph $G = (V, E)$ is defined as the list of eigenvalues (together with their multiplicities) of its adjacency matrix $\mathbf{A}$; even nonisomorphic graphs can share the same spectrum [128]. It is often more convenient to study the eigenvalues of the *Laplacian* matrix ∇ of G instead of $\mathbf{A}$. This is defined element-wise as (see e.g. [27])

$$\nabla_{uv} = \begin{cases} 1, & \text{if } u = v \text{ and } \deg(v) > 0, \\ -\dfrac{1}{\sqrt{\deg(u) \cdot \deg(v)}}, & \text{if } u \in \Gamma(v) \\ 0, & \text{otherwise.} \end{cases} \tag{2.3}$$

If $\mathbf{D}$ is the diagonal degree matrix $\mathbf{D}_{vv} = \deg(v)$, the adjacency matrix and the Laplacian matrix are related by the following equality:

$$\nabla = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}, \tag{2.4}$$

where $\mathbf{D}_{vv}^{-1} = 0$ if $\deg(v) = 0$. This is convenient as the eigenvalues of ∇ all fall within the interval $[0, 2]$, the smallest eigenvalue always being zero, and therefore the spectra of different graphs of possibly very different orders become more comparable [27, 128].

A *family* $\mathcal{G}$, also called an *ensemble*, of graphs includes all graphs that fulfill the family description; the description usually contains parameters that control the size of the family. One commonly used family is $\mathcal{G}_n$, which contains all graphs of order n. A *property* $\mathcal{P}$ of graphs in the ensemble $\mathcal{G}$, as defined in [19], is a closed subset of $\mathcal{G}$ such that if $G \in \mathcal{P}$ and $G' \in \mathcal{G}$, then $G \cong G'$ implies that $G' \in \mathcal{P}$. A property $\mathcal{P}$ is *monotone* if $G \in \mathcal{P}$ and $G \subset G'$ imply $G' \in \mathcal{P}$, and *convex* if $G'' \subset G' \subset G$ and $G'', G \in \mathcal{P}$ imply that $G' \in \mathcal{P}$. When $G \in \mathcal{P}$, we say that G *has* property $\mathcal{P}$.

2.3 Case study 1: Modeling the Internet

The Internet has been an object of intensive study ever since its economical value was recognized in the 1990s. Paxson and Floyd [115] found in 1997 the topology of the Internet difficult to characterize due to the constant change and growth that are essential qualities of the network. They motivate the research on Internet simulation by the possibility to approach "complicated scenarios that would be either difficult or impossible to analyze" [115]. Just a few years later the size of the network has exploded and the problems related to its topology are more urgent than ever; more efficient protocols are needed for the ever-growing amount of traffic. It would also be helpful to be able to predict the future evolution of the network in order to design better hardware, traffic protocols, and routing algorithms [46, 128].

The methods for generating "Internet-like" networks can be classified into three groups [92]:

- **Random networks**: Fix a set of nodes on some space such as a coordinate grid, usually located randomly according to the uniform distribution. Add connections between the nodes with a probability that follows some meaningful formula, for example related to the distance between two nodes.
- **Regular networks**: Pick some regular structure, such as a grid, on which the nodes are placed. Add connections according to the regular structure following some (deterministic) rule.
- **Hierarchical networks**: Start by building a set of smaller graphs by some generation method and connect these into the "next level" of the hierarchy following a meaningful rule. Continue for as many steps as appropriate.

Any successful method would have to be a combination of these, as it is quite obvious that the Internet is neither completely regular nor entirely random; both elements are present. Also, by observing the growth mechanism and loose control of the Internet, some hierarchical element is surely present: computers are organized first into local networks that form domains; domains are further connected to form a larger structure, and so forth. In this section we take a glance at some models that have been proposed to model the Internet and the observations made during these modeling efforts. Many of the phenomena present in this case study will also appear later in the theoretical discussions on generating realistic networks in Chapter 3.

2.3.1 The Waxman model and its variants

In 1988, Bernard M. Waxman [135] presented a model for the Internet with the intent to examine the routing of multipoint connections. He simplified the network into a graph $G = (V, E)$, where vertices represent the switches and edges are the links between them. To make his model realistic, Waxman assigned a capacity to each edge that represents the bandwidth of the actual connection and also a cost to be used as a weight factor in calculating the "real" path lengths, which are in reality more informative than the plain distances $d\,(u, v)$. The problem of *multipoint* (commonly *multicast*) routing, to which the model was tailored, is a problem of finding a subtree of $G = (V, E)$ with the smallest total edge cost that reaches all vertices in a given vertex set $S \subseteq V$ (i.e., a minimum *Steiner tree*).

Waxman [135] described two random graph models that capture some characteristics of real networks: seemingly random connections that are however sensitive to physical separation of the nodes. In the first model, n vertices are randomly and uniformly distributed over a rectangular grid and labeled with the corresponding coordinates. The distance $dist_E(u, v)$ between each pair of vertices $u, v \in V$ is then calculated as the Euclidean distance on the coordinate grid. Edges are placed between pairs of vertices $\{u, v\}$ with a probability P that depends on $dist_E(u, v)$ and two constant parameters $\alpha, \beta \in (0, 1]$. The maximum distance on the grid is denoted by L.

$$\Pr\left[(u, v) \in E\right] = \alpha \exp \frac{-dist_E(u, v)}{\beta L}. \tag{2.5}$$

Waxman does not state why he chose this form of function which is common in statistical mechanics (see e.g. [77]), but we expect the justification to be that the probability decreases exponentially with distance, is relative to the maximum distance, and can be scaled with the parameters; α controls the density of the graph, and β controls the relative density of edges with low and high cost, where the cost of an edge (u, v) is set to be $dist_E(u, v)$.

In Waxman's second model, the distances are not the Euclidean distances on a grid, but are randomly chosen from a uniform distribution on a given interval $(0, L)$. The edges are then placed as in the former model. Waxman does not explain how the graphs generated by these procedures resemble or differ from the Internet; his experiments were of small scale in comparison to those performed nowadays; n was fixed to 25 and only 5 graphs were generated and simulated on for both models. However, the Waxman model became and remained very popular until recently, when more complex models began to emerge [39, 46, 138].

In 1996, Matthew Doar [39] argued that although the Waxman models have been commonly used in generating graphs, they are not sufficient. He proposed modifications to the coordinate grid model and studied the properties of graphs generated by the modified procedure. Doar took a layered approach and modeled the Internet as a set of smaller networks with intermediate connections. The first layer to be generated is a set of Local Area Networks (LAN), which are then connected into Autonomous Systems (these are the Internet domains), which can be connected further to form larger entities. The parameters of the model are the numbers of "network units" on each layer and the degree of vertices for each layer separately, as well as the number of edges connecting the layers into a single graph. The algorithmic description of the model relies on the use of a coordinate grid as in the Waxman models, as well as in Doar's earlier modification of the Waxman model together with Ian Leslie in 1993 [38].

In 1997, Zegura, Calvert and Donahoo [138] suggested and examined two direct modifications to the edge distribution of the Waxman models: the "exponential method" (Equation 2.6) and the "locality method" (Equation 2.7), replacing Equation 2.5 respectively by the following definitions:

$$\Pr\left[(u,v) \in E\right] = \alpha \exp \frac{-d\,(u,v)}{L - d\,(u,v)}, \tag{2.6}$$

$$\Pr\left[(u,v) \in E\right] = \begin{cases} \alpha \text{ if } d\,(u,v) < \delta \\ \beta \text{ if } d\,(u,v) \geq \delta, \end{cases} \tag{2.7}$$

with $\alpha, \beta \in (0, 1]$ and δ is a constant. They discuss and experiment on the values that should be assigned to the parameters to produce interesting networks with these distributions or the original distribution by Waxman.

Zegura et al. [138] also present a hierarchical generation method somewhat similar to Doar's layered model [39], and another hierarchical method they call the *Transit-Stub method* that uses random models (either Waxman's or their own) to produce the building blocks of the hierarchical structure under construction. They concentrate on the metrics used to describe a network, such as the average degree of vertices, the diameter of graph and the number of biconnected components [137, 138]. They also analyze the expected value of the vertex degree in the Waxman model and their own exponential method in [138], and for the Transit-Stub method in [137] by Zegura, Calvert and Bhattacharjee. Such analysis together with the use of statistical methods to analyze the resulting networks indicates that they were in part leading the way to current research practices.

Calvert, Doar and Zegura [25] present a detailed outline of the generation of layered Internet-like graphs. They discuss two different implementations of the described model: the Transit-Stub model of [137] briefly described above and another implementation called *Tiers*, which is essentially the model of Doar [39] also explained above in general terms. A brief note on the practicalities of choosing a proper generator for some specific application is provided in [25].

2.3.2 Power-law behavior on the Internet

Recently the research effort on Internet topologies has been tremendous and several researchers have observed *power law* distributions, also known as *heavy-tail* distributions, for instance in the growth of the World Wide Web and the Internet itself [23, 46]. A power law is simply an expression of the form $x \propto y^{\beta}$, where x and y are the measured values and β is "near-constant" [46]. More formally, a nonnegative random variable X obeys a power-law distribution if

$$\Pr\left[X \geq x\right] \sim \alpha x^{-\beta} \tag{2.8}$$

for constants $\alpha, \beta > 0$. In some cases also a slowly varying function[7] $f(x)$ is included as a coefficient of the probability [46]. Power laws have been discussed as early as 1955 by Simon (and even earlier by Vilfredo Pareto; see [121] and the references therein) in the context of for example income distributions, city sizes, and the number of authors contributing to scientific publications. The last example remains topical and will be addressed in Section 5.3. Therefore models for random graphs with a power law degree distribution have been suggested and studied intensively (see for example [13, 23, 80]).

Michalis, Petros and Christos Faloutsos [46] have studied Internet topology in order to enable design of better protocols and more accurate simulation. They reproach the need for intuition and experimental work in choosing proper values for the parameters in graph generation models such as those presented above in Section 2.3.1. They are also discontent with the metrics used to characterize graphs, which are mainly measures of degree and distance distribution. Especially average values are not very informative for a power law distribution, they argue, proposing definitions of their own to better characterize networks, including the power-law exponents of Equation 2.9, defined for an undirected graph $G = (V, E)$.

$$\begin{array}{rclcrcl} \deg(v) & \propto & \mathrm{rank}(v)^{\mathcal{R}} & \qquad & f_d & \propto & d^{\mathcal{D}} \\ P(h) & \propto & h^{\mathcal{H}} & & \lambda_i & \propto & i^{\mathcal{E}} \end{array} \tag{2.9}$$

- $\mathcal{R}$ is the *rank exponent*. The *rank* of a vertex is the index of that vertex in order of decreasing degree.
- $\mathcal{D}$ is the *degree exponent*, which in [46] is called the *out-degree exponent*, but corresponds to the total degree $\deg(v)$ for an undirected graph. $f_d = |\{v \mid v \in V,\ \deg(v) = d\}|$ denotes the number of vertices with degree d.
- $\mathcal{H}$ is the *hop-plot exponent*. $P(h) = |\{\{u, v\} \mid u, v, \in V,\ dist(u, v) \leq h\}|$ denotes the number of vertex pairs that are within h "hops" of each other in $G = (V, E)$. This is meaningful only for values of h that are significantly smaller than $\mathrm{diam}(G)$.
- $\mathcal{E}$ is the *eigenexponent* that characterizes the spectrum of G, consisting of eigenvalues λ_i of order i.

For the Internet, $\mathcal{H} > 0$ whereas the other three are positive. These power-laws are intended for characterization of graph topologies into "realistic" and "artificial" graphs instead of mere average values of more traditional metrics, and have been enthusiastically accepted as indicators of Internet-like topology (see for example [24, 70, 92, 94]). Generation models based on these observations are described in the next section to portray the current state of Internet modeling.

[7] A positive and measurable function $f(x)$ is *slowly varying* if $\forall t > 0\ f(tx) \sim f(x)$ as $x \to \infty$.

2.3.3 Modern generation models

Medina, Matta and Byers [92] propose the BRITE (Boston University Representative Internet Topology Generator), based on an observation by Barabási and Albert [12] that for a power law to be present in a network, the network construction needs to exhibit *growth* and *preferential attachment*. This means that the number of nodes in the network grows in time and the new nodes will form connections to the old ones based on the number of connections each old node has gathered thus far; the more connected an old node is, the more "popular" it is to connect there. Nodes that have many connections are therefore more likely to attract new connections than nodes with low initial degree. We return to these foundations of the BRITE model in Section 3.3.1.

The generation procedure of BRITE is founded on an $\mathcal{H} \times \mathcal{H}$ grid of high-level squares, each divided further into a grid of $\mathcal{L} \times \mathcal{L}$ low-level squares. For each high-level square, a random number of nodes are placed. The probability distribution for the number of nodes is chosen as the *bounded Pareto distribution*, *Pareto* (k, p, α), where $-(\alpha + 1)$ becomes the power-law exponent [32]

$$f(x) = \frac{\alpha k^{\alpha}}{1 - (k/p)^{\alpha}}\, x^{-(\alpha+1)}, \quad k \leq x \leq p. \tag{2.10}$$

At most one node can be placed in each of the low-level squares and therefore $x \leq \mathcal{L}^2$ must apply. The nodes are placed within their respective high-level squares randomly and uniformly, avoiding collisions: if the square being filled is already occupied, draw another random low-level square. Nodes are assigned d connections as they are positioned. If no incremental growth is desired, all nodes are positioned simultaneously and then connected so that each connects itself to d nodes selected from all possible nodes as described below.

In the incremental version, a node may only connect to nodes that are already present in the network when the node itself gets placed. Initially, a small subset of at least d nodes are placed and randomly connected so that incremental connection procedure can properly continue. The method of choosing the connections can be varied: one may choose either the Waxman probability function (Equation 2.5), or a preferential probability relative to the node degree (essentially Equation 3.25 on page 47), or a combination of these.

Medina et al. [92] also study some recently proposed metrics of network models for different Internet topology generators: the presence of power laws (as defined by Faloutsos et al. [46] discussed in the previous section), the average path length between pairs of nodes, and the average density of subgraphs induced by vertex neighborhoods, which is a clustering property. They name the following four factors that they believe to significantly affect the behavior of a generator:

1. **Incremental growth of the network:** New nodes can be introduced to the network and connected one by one following some specified schedule.

2. **Preferentiality of connections:** Is a newly introduced node more likely to connect to nodes that already have a large number of connections?

3. **Geographical distribution of nodes:** How far will the nodes be from each other physically, i.e., what are the typical distances between nodes?

4. **Locality of the connections between nodes:** Does a newly introduced node rather connect to nodes that are physically close than to nodes located further away?

The topology generators examined by Medina et al. are the Waxman model [135] and the Transit-Stub model [25, 39] of Section 2.3.1, regular grids, and of course, BRITE itself. They find that of the four power laws of Equation 2.9, the rank power law with $\mathcal{R}$ and the degree power law with $\mathcal{D}$ "are most effective in distinguishing different kinds of topologies", and that even though the hop-plot power law and the eigenvalue power law of Equation 2.9 apply for all the tested generators, the values of the exponents $\mathcal{H}$ and $\mathcal{E}$ vary [92]. They believe, in agreement with Barabási and Albert [12], that preferential attachment and incremental growth are truly responsible for the presence of such power laws. They also find that incremental growth seems to increase both the average path length and the "clustering" of connections into dense neighborhoods.

The Inet generator [70] by Jin, Chen, and Jamin takes three parameters: the order of the graph, the fraction of vertices with $\deg(v) = 1$, and the size of the plane on which the vertices are placed to simulate physical nearness. The order of generated graphs is recommended to exceed 3,037, which is the number of ASs on the Internet in November 1997, used as the foundation of the generator design. They compare their generator to four other generators, including BRITE and the approaches of Waxman and Doar described above and an earlier version of Inet itself. They study the how the generated graphs follow the power laws observed on the Internet, but report their results narrowly, comparing a single snapshot of the Internet to only five randomly generated graphs with somewhat unconvincing argumentation.

The field appears to be open for better and better Internet topology generators, although it is already difficult to determine whether one generator outperforms the other, due to the abundance of possible metrics and the ever-changing measurement results on the current state of the Internet.

Mihail et al. [94] study the clustering properties (see Section 4.5) of the graphs generated by Internet topology generators. They examine graphs of order 11,000, two generated with BRITE and three with Inet (which is not a very large sample), isolating the 2,200 vertices with the highest degree to locate the core. They compute the first 30 eigenvalues from the subgraphs induced by these "core" vertices and use these to find clusters.

In their experiments, Mihail et al. note that the clustering coefficient $\mathcal{C}$ is not a proper measure of clustering as Inet matches quite well the value of $\mathcal{C}$ of real Internet data, but differs significantly in spectral properties. This is particularly true with respect to the clusters found by their own method as they study the spectrum of the graph. They conclude that the clustering produced by degree-sequence dependent generation methods is weak in comparison to that of real data. Vukadinović et al. [128] aim to classify natural graphs and graph generators with spectral methods, studying the multiplicity of eigenvalue one. They compare the classification results of a domain-level Internet graph to that of the Inet generator.

2.4 Case study 2: Modeling the World Wide Web

Another widely studied network is the World Wide Web, with either individual pages or entire websites as vertices, and links as edges. The interest in mathematical models is justified by the size of the resulting network; in 1999 even the best search engines could cover only about one third of the estimated size of the WWW [87]. Since then, the network has grown significantly and full indexing is impossible. In this section we review the models proposed and some of the key observations made concerning the structure of the World Wide Web. For a survey on metrics related to the WWW we direct the reader to [36] by Dhyani et al.

The routing graph of the Internet is generally considered undirected, but hyperlinks are most obviously directed and therefore the in-degree and out-degree of vertices should be handled separately instead of simply looking at the total degree of a vertex. The in-degree represents in a sense the *popularity* of a website $v \in V$: how many administrators of other websites u_i have chosen to put a hyperlink $\langle u_i, v\rangle$ on their website leading to v. The out-degree in turn represents the *forward connectivity* of a website: how many hyperlinks has the administrator of $v \in V$ decided to put on his site pointing to other sites w_i by hyperlinks $\langle v, w_i\rangle$.

One starting point of the World Wide Web modeling was a paper by Kleinberg et al. [80] in 1999, in which the authors construct a graph to represent the WWW. We adopt their practice of calling such a graph the *Web graph*. Their motivation is improving search performance on the WWW as well as providing more accurate topic-classification algorithms; they foresee a "growing commercial interest" in WWW modeling, which is still a major driving force in network research.

Broder et al. [23] were the first to map the structure of the Web graph. They confirm that the degree distributions follow power laws, but the main contribution is the study of the connected components, both directed and undirected, of the

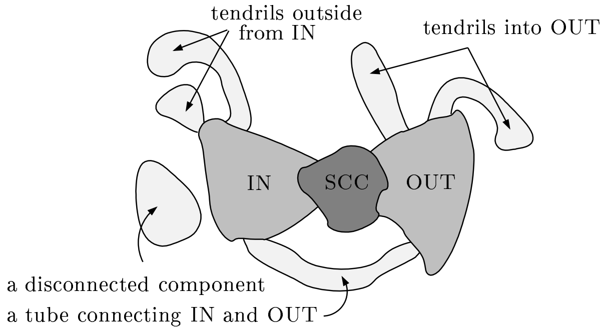


FIGURE 2.3: A structural diagram of the WWW (adapted from [23]); this is often called the "bow-tie diagram".

Web graph. The largest strongly connected component (SCC) of the graph — i.e., a subgraph in which any page can be reached by hyperlinks from any other page — they call the *central core* of the graph. The next two major subgraphs are called IN and OUT, where there exists a sequence of hyperlinks connecting each page in IN to the central core but not vice versa, and respectively OUT contains those pages that can be reached from the central core but do not have hyperlinks pointing back at the SCC. The rest of the Web graph Broder et al. call the *tendrils* of the WWW; it consists of pages that cannot reach the central core or vice versa. See Figure 2.3 for illustration.

By performing and analyzing three webcrawls[8] that span approximately 200 million webpages, they found that the diameter of the central core is at least 28 and that of the entire Web graph as seen in the year 2000 at least 500. The order of the SCC was approximately 56 million pages whereas the IN, OUT and the tendrils each contained approximately 44 million pages. The giant connected component in the undirected sense that contains all the pages connected to the SCC, IN, or OUT in either direction is quite large, also in comparison to the size of the SCC, containing 186 million pages.

The calculation of the diameters, 28 for the SCC and 500 for the giant undirected component, has naturally not been exhaustive over all pages in the components but based on a small sample of starting points for a breadth-first analysis; the SCC sample contained 136 vertices, the IN sample 128 vertices, and the OUT sample 134 vertices. The orders of the components were approximated by analyzing breadth-first searches from 570 random starting points.

Huberman and Adamic [66] have found that the number of webpages per website

[8] In a webcrawl, a software agent follows hyperlinks proceeding from page to page automatically, reporting the structure of the "scanned" network. The crawling often proceeds *breadth-first*, following all the links on one page before using the links on the subsequent pages.

follows a power law distribution. Albert, Jeong and Barabási [6] have studied the Web graph finding power laws for both the in-degree distribution and the out-degree distribution, as well as a fairly small average diameter: on average 19 links suffice to navigate from a webpage to any other arbitrarily chosen page. They discuss with Barabási et al. in [4] the webcrawl performed by the latter in [12] that displays characteristics of a power law in the link distribution of the World Wide Web. Barabási et al. have proposed earlier in [12] that the number of links increases with the age of the website (i.e., the longer the lifetime, the larger the number of incoming links). Adamic and Huberman [4] argue that the age of a webpage does not correlate with the amount of links it has gained, but only its initial 'attractivity': "all sites are not created equal". Both approaches seem to be useful in current models of the WWW: nodes are assigned an initial attractivity value, but also their lifetime influences the number of links they will attract.

Broder et al. [23] examined the degree distributions of their sample of the Web graph and verified the presence of power laws for the in-degree and the out-degree of the Web graph, originally reported for experiments of smaller scale such as that of Barabási and Albert [12]. They report $\gamma_{\text{in}} = 2.09$ which is in agreement with [12], and $\gamma_{\text{out}} = 2.72$. Many papers have been written on the origin of these power laws in the growth process of the WWW (see for example [88, 124, 126]), but we have not yet encountered recent measurements of similar scale. However, sampling the World Wide Web reliably is nontrivial [116]; the first problem is to define a "random" webpage, and the second to construct a method to obtain one. Taking a "snapshot" of the Web graph by some sort of a webcrawl is slow if a large portion of the network is mapped. The WWW is continuously changing and therefore the parts of the network covered by early phases of the crawl represent in fact a different version of the graph than the parts covered later. As the time span may be days or even weeks, this cannot be overlooked. However, webcrawls are currently the best means to obtain such information. Reliable sampling of the WWW and other large networks will be a pressing task in further work on this area.

Kleinberg et al. [80] observed two special classes of webpages in relation to a certain topic in the content of the pages: *authoritative* pages that are devoted to the topic, and *hub* pages that contain several links to other pages related to the same topic. This observation was used to improve search algorithms. While conducting experiments on these algorithms, Kleinberg et al. made several measurements of the Web graph. They compare the frequency of vertices with the same in-degree on a log-log plot of $\deg_{\text{in}} \in [0, \Delta_{\text{in}}]$ versus the number of vertices $[0, n]$ with each value of $\deg_{\text{in}} z$. They noted that $\Pr[\deg_{\text{in}}(v) = k] \sim 1/k^{\alpha}$, $\alpha \approx 2$. Such distributions are called *Zipfian* (due to [140]) and differ from those predicted by previous network models. A Zipfian distribution is in fact such that the log-log plot of the *ranks* of the values of the random variable versus their frequency follows a straight line. In a power-law distribution, the values themselves versus their frequencies on a log-log plot fall on a line. The out-degree log-log plot shows a similar distribution than the in-degree, although not as clearly.

Kleinberg et al. [80] stated the obvious question: if existing models of random graphs do not explain this behavior, "what is a natural stochastic process that will?" They believe that such a model would allow for better modeling of the WWW, more informative predictions on the efficiency of algorithms — especially those with poor worst-case behavior — as well as predicting future development of the structure of the WWW. As a solution, they suggest generating networks using random *copying* of neighborhoods from existing vertices to newly added vertices: from an existing vertex $v \in V$, some neighbors $S \subseteq \Gamma(v)$ are chosen to form (a part of) the neighborhood of a new vertex $u \in V$.

The model of Kleinberg et al. contains four discrete-time stochastic processes, one that creates vertices, one that creates edges, one that deletes vertices, and one that deletes edges. The vertex processes are simple: at time t, create a new vertex with probability $\alpha_c(t)$. The vertex deletion is a Bernoulli process with deletion probability $\alpha_d(t)$; as a vertex is deleted, all of the adjacent edges are also removed. The edge deletion process of the model removes a vertex v_t with probability δ. The authors however point out that an ideal deletion process would have δ nonincreasing in $\deg_{\text{in},t}(v_t)$; we find it somewhat confusing that edges are not deleted without deleting vertices. In the process of edge creation at time t, a vertex v_t is chosen from the graph according to a probability distribution. This vertex v will be the source of the created edges (v_t, u_i), $i \in \{1, \ldots, k\}$, where k is sampled from a given probability distribution. Therefore, $\deg_{\text{out},t+1}(v_t) = \deg_{\text{out},t}(v_t) + k$. The k edges are created as follows:[9]

- With probability β, the target nodes of edges (v_t, u) are chosen uniformly and independently at random from the graph G_t.
- With probability $1 - \beta$, a vertex u, $\deg_{\text{out}}(u) = h$ is chosen randomly and k vertices from $\Gamma(u)$ are chosen as neighbors of v_t in G_{t+1}. If $h < k$, another node $u' \in V$ is randomly chosen and the remaining $k - h$ neighbors are copied from u'. This is repeated until k edges have been obtained for v_t.

To gain some intuition about this model, consider the publication of a new webpage on some topic with some links to other pages. Some of these links are likely to be the same that are listed on other pages on that topic, such as major organizations related to the topic. This part of the neighborhood structure can be considered as "copied" from some previously existing webpage. In addition, the author of the new page will possibly like to contribute another point of view to the topic and is likely to include some "fresh" links on the page in addition to the "established" links

[9] The authors do not state whether multiple edges between a pair of vertices are allowed or omitted in the edge creation process.

on the topic. New webpages appear and some are removed; both processes seem random to the outside observer. The number of links on a page is not constant in real life and therefore it is drawn from a properly chosen distribution in the model.

Kumar et al. [84] propose a family of stochastic models for the Web graph that grow and change structure in time. The growth may be either linear or exponential and the edges are placed with a copying process similar to that of Kleinberg et al. [80] above. Another option for introducing the links is at uniform, choosing the endpoints independently at random from a growing graph. As the goal is to model the World Wide Web, the generated graphs are directed.

In the linear growth copying model, a newly added vertex may form a directed link to any other vertex existing at that time. Kumar et al. fix a constant out-degree d for the added vertices and a *copy factor* $\alpha \in (0, 1)$. One vertex v is added per time step and the d edges are either chosen uniformly with probability α or copied from an already existing vertex w so that if the ith edge of w with respect to some fixed edge ordering is $\langle w, u\rangle$, the ith edge of v is $\langle v, u\rangle$. The vertex w will remain fixed during the edge-creating process of vertex v; intuitively, in practical terms, w represents an established webpage on the same topic as a newly created webpage represented by v and has been chosen as a prototype by the author of the new webpage.

The exponential growth copying model has more parameters: the rate of growth, out-degree, the copy factor, and a self-loop factor. A newly added page (as there are now several of them, more at each time step due to the exponential growth) may only point to pages that existed before the current time step and not to those that are being created at the same time. Kumar et al. [84] also suggest introducing a death process for both vertices and edges as a future generalization of these models. Also the selection of the prototype vertex and the edges pointing out of the new vertices could be modified to produce a desired power law. They derive the degree distributions and bounds for the number of cliques in the generated graphs.

The design of stochastic models for the Web graph was continued by Levene et al. [88] who aim to match the data measurements reported by Broder et al. in [23], where $\gamma \approx 2.1$. This is achieved by combining a preferential attachment process with a nonpreferential one, building on the foundation of Simon's early model [121] and considering the process as a *urn transfer model*. Assume initially that there is a countable number of urns where balls can be placed. The urns are labeled with the integers $i = 1, 2, 3, \ldots$ and each ball in urn u_i has exactly i pins attached to it. The process will be discrete-time and has two parameters: $\alpha > -1$ and $p \in (0, 1)$. Initially at time $n = 1$, urn u_1 contains one ball and the other urns are empty. At time $n \geq 1$, a new ball with one pin is added to urn u_1 with probability

$$p_{n+1} = 1 - \frac{(1-p)\sum_{i=1}^{n}(i+\alpha)F_i(n)}{k(1+\alpha p)+\alpha(1-p)}, \tag{2.11}$$

where $F_i(n)$ is the number of balls in urn u_i at time n by $F_i(n)$. Note that $\mathrm{E}[p_{n+1}] = p$. With probability $1 - p_{n+1}$, and whenever $p_{n+1} \notin [0, 1]$, one ball from u_i is transferred to urn u_{i+1} with an additional pin — the urn u_i is selected randomly with probability that depends on the number of balls in u_i:

$$\Pr[u_i \text{ is chosen}] = \frac{(1-p)(i+\alpha)F_i(n)}{k(1+\alpha p) + \alpha(1-p)}. \tag{2.12}$$

Note that an empty urn is never chosen. At each time step, exactly one new pin appears, either along the new ball inserted to the first urn or as a result of transferring a ball one urn up. Therefore at time n, there are exactly n pins in total in all the balls in all of the urns. For $\alpha = 0$, the transfer of a ball from an urn is purely preferential as $\Pr[u_i \text{ is chosen}] = \frac{1-p}{k} i F_i(n)$. Larger values of α introduce a nonpreferential component in the selection process. Denoting $\lim_{n\to\infty}(\frac{1}{k}\,\mathrm{E}[F_i(n)]) = f_i$, Levene et al. [88] derive that asymptotically $f_i \sim c i^{-1(1+\rho)}$, where c is a constant independent of i and $\rho = (1 + \alpha p)/(1 - p)$. Hence they may control the exponent of the power law by adjusting the parameters α and p.

In terms of the Web graph, adding a new ball corresponds to the creation of a webpage with one incoming link, and moving a ball to the next urn corresponds to adding a new incoming link (the level of preferentiality depending on α) to an existing webpage. As the average in-degree of a webpage is reported to be approximately 8 in [85], yielding $p = 0.125$, and the power-law exponent to be 2.09 in [23], Kumar et al. [88] arrive at $\alpha \approx -0.37$. Looking at the out-degrees, where the average is approximately 7.2 links per page according to [85], $p = 0.14$ and hence $\alpha \approx 3.42$. Kumar et al. also provide other interpretations related to the properties of the Web graph along with simulation results. Their concluding hypothesis is that the the evolution of the Web graph cannot be explained by preferential processes alone.

Chapter 3

Mathematical network models

Alongside with empirical work, mathematical modeling of natural phenomena has been a major tool for scientific discovery. The task of choosing a proper description and a sufficient level of detail is nontrivial. As seen in the previous chapter, network models have been employed to study various phenomena where connections between certain entities are of interest. They may be either *deterministic*, such as the network models built to match a given set of data (for example the neural network of *C. elegans* of Section 2.1), or *stochastic* (such as the Internet models of Section 2.3). This chapter emphasizes the latter, but a couple of deterministic models are presented as well. The goal of a stochastic network model is to produce a network with *similar characteristics* as the modeled phenomenon. This approach is often applied when complete network data for the phenomenon is unavailable or only represents a small sample of the population being modeled.

In this chapter we discuss three main classes of network models: the traditional approach of uniform random graphs, and the recent suggestions of *small-world* and *scale-free* random graphs. Variations of all three are presented and their properties discussed. The first section addresses the models that consider all edges equiprobable; for the first two approaches, the uniform model and the percolation model, each vertex will receive the same expected number of edges. The third approach allows for a predetermined degree distribution that the graph must meet.

The genre of small-world networks differs from this by combining randomness and structure; the common generation models first create a regular graph over which random edge replacement or addition will take place. Another approach is taken to generate scale-free networks: scale-free graphs can be grown by adding new vertices to a small graph and connecting the new vertices to existing vertices. The goal of this model is to mimic natural processes in the connection procedure and obtain a scale-free degree distribution similar to that obtained by the natural process. Some approaches that aim to combine elements from the small-world models and the scale-free models are also discussed. We conclude the chapter with some *deterministic* generation models that do not involve a random element.

3.1 Random graphs

In algorithm analysis, complexity has traditionally been evaluated over all possible inputs. It is acknowledged that only studying the properties of an algorithm for the *worst-case* input is not sufficiently informative; it would be desirable to know how the algorithm behaves on an *average* instance (see e.g. [119]). The problem is to determine the *distribution* from which the input instances are drawn. Network modeling shares the same goal: what does a *typical* network look like and what properties can it be assumed to possess?

The standard answer has been for decades that a *randomly* generated network will be such an "average instance" and that the properties of random networks will lead to greater understanding of the application problems at hand. When designing an algorithm for a particular application, such as routing on the Internet, a "typical" instance of a communication network is hardly random, as seen in the previous chapter. It would be necessary to obtain a random "Internet-like" graph to make justified and practical statements about the efficiency of a particular routing algorithm designed to operate on the Internet.

We begin this section by presenting the traditional view of random graphs according to which a particular connection between two vertices will be as likely to form as any other connection, all connections being pairwise independent. We then move on to two recent proposals for models of random graphs that match some real-world application problems better than the uniform random graphs. Both of these proposals, the *small-world* network model initiated by Watts and Strogatz [134] and the *scale-free* network model initiated by Barabási and Albert [12], emphasize that there is some structure in the connection topology that is not uniform. Some nodes of natural networks are just more important than others, as well as some connections have a stronger influence on the network structure than others.

3.1.1 The Erdős-Rényi model: Uniform random graphs

This section describes two standard models of uniform random graphs that have been intensively studied for the past decades. Both models produce graphs in which the number of vertices is fixed and all edges have the same probability to appear in the graph. A standard reference on such graphs is Béla Bollobás' book "Random Graphs" [19].

In 1959, E. N. Gilbert [54] presented a process for generating random graphs with

n vertices: each of the $\binom{n}{2}$ possible edges is included in the graph with probability p, considering each pair of vertices $\{v, w\}$ independently. The family $\mathcal{G}_{n,p}$ of these graphs contains $2^{\binom{n}{2}}$ possible graphs in total. An instance of the family $\mathcal{G}_{n,p}$ is often denoted by $G_{n,p}$. Such a process creates any $G \in \mathcal{G}_{n,p}$ with equal probability; an equivalent method would be removing each edge from the complete graph K_n with probability $1-p$ independently of the other edges. Gilbert studied the probability of a $G \in \mathcal{G}_{n,p}$ being connected. This can be expressed in terms of the number of connected graphs with $|V| = n$ and $|E| = m$, denoted $C(n, m)$, as each such graph has probability $p^m(1-p)^{\binom{n}{2}-m}$ of being chosen:

$$\Pr\left[G \text{ connected}\right] = \sum_{m=n-1}^{\binom{n}{2}} C(n, m) p^m (1-p)^{\binom{n}{2}-m}. \tag{3.1}$$

The main results of this early paper presenting a now common model of random graphs were the following bounds for a random graph G with n vertices and a random pair of vertices $\{u, v\}$:

$$\begin{aligned} \Pr\left[G \text{ is connected}\right] &\sim 1 - n(1-p)^{n-1} \\ \Pr\left[d\left(u,v\right) < \infty\right] &\sim 1 - 2(1-p)^{n-1}. \end{aligned} \tag{3.2}$$

In the $G_{n,m}$ model proposed by Erdős and Rényi [43, 44], the number of vertices is again fixed to n, but instead of including each edge independently with probability p, a total of m edges are randomly drawn from the set of all possible $n(n-1)/2$ edges. In [43] Erdős and Rényi study the probability that $G \in \mathcal{G}_{n,m}$ is connected, together with other probabilities related to the connected components of a graph. In [44] they study the structure of a "typical" graph in $\mathcal{G}_{n,m}$ when n grows to infinity and $m = m(n)$, determining several *threshold functions* to characterize what properties a typical instance is likely to have.

The $G_{n,p}$ and $G_{n,m}$ models are in many ways equivalent (see for example [19] for a detailed explanation of the small differences); literature often credits both to Erdős and Rényi. When referring to a random graph, we generally mean a member of $\mathcal{G}_{n,p}$, using the acronym ER for these traditional uniform models of random graphs. Note that all graphs in $\mathcal{G}_{n,m}$ are included in $\mathcal{G}_{n,\delta}$ for $\delta = 2m/n$, as the connection probability p is equivalent to graph density. The probability spaces of the two families $\mathcal{G}_{n,p}$ and $\mathcal{G}_{n,m}$ are hence somewhat different, but all the interesting properties are essentially identical.

We mention here two situations where $\mathcal{G}_{n,m}$ can be replaced with $\mathcal{G}_{n,p}$ without losing significant information [19]. First, if a sum of expectations of a random variable X, $\sum_{m=0}^{\binom{n}{2}} \mathrm{E}_m(X)$ is concerned, then no assumptions on X are needed to exchange between the models. The second situation is related to the properties of the families. We say that *almost every* graph G in a family $\mathcal{G}_n$ of random graphs of order n has a property $\mathcal{P}$, if $\Pr\left[G \text{ has } \mathcal{P}\right]$ goes to one as $n \to \infty$. If almost every graph $G \in \mathcal{G}_{n,p}$ has a convex[1] property $\mathcal{P}$, then almost every graph $G \in \mathcal{G}_{n,m}$ where $m = \lfloor pn \rfloor$ also has $\mathcal{P}$.

Both of the models $G_{n,p}$ and $G_{n,m}$ can also be described as *random graph processes* [89]. A random process is a *stochastic process* $\{X_n,\ n = 0, 1, 2, \ldots\}$ where the value of X_i is the *state* of the process at time i (continuous-time processes are also possible). The process is defined through the set of possible states and *transition probabilities* between the states. In a random graph process, each state characterizes a graph and the transitions introduce modifications to the graph, such as the addition of an edge. A good textbook on stochastic processes is [62] by Grimmett and Stirzaker.

Consider a stochastic process $\{G_{n,m}\}_{m=0}^{\binom{n}{2}}$ starting on an empty graph of n vertices, and "growing" from $G_{n,m-1}$ to $G_{n,m}$ through the addition of a new edge at time m uniformly at random among the $\binom{n}{2} - m + 1$ possibilities. For the process $\{G_{n,p}\}_{p\in[0,1]}$, consider a family of independent random variables $\{X_{i,j}\}_{i,j\in V}$ uniformly distributed over $[0, 1]$. Now the edge set of a $G_{n,p}$ is obtained as $E = \{(i, j) \mid X_{i,j} < p\}$.

The degree of a single vertex of $G \in \mathcal{G}_{n,p}$ follows the binomial distribution: $\deg(v) \sim \mathit{Binom}(n-1, p)$, from which it follows that for a random variable X_k representing the number of vertices with degree k, it applies that X_k is asymptotically Poisson distributed (see for example [19]):

$$\Pr\left[X_k = r\right] \sim \mathit{Poisson}\left(\lambda_k\right) = \frac{\lambda_k^r}{r!}\, e^{-\lambda_k}, \tag{3.3}$$

where $\lambda_k = \binom{n-1}{k} p^k (1-p)^{(n-1)-k}$ and r is any fixed integer. Also $\mathrm{E}[\delta] = p$, as $\mathrm{E}[m] = p\binom{n}{2}$ and $\delta = m/\binom{n}{2}$, and the density of any subgraph of order h has expected value p, as within the subgraph, each of the $\binom{h}{2}$ edges is included with probability p.

3.1.2 Percolation

In this section we briefly review the concept of percolation for further reference. For a thorough view on percolation, we recommend the work of Geoffrey Grimmett, see e.g. [60, 61]. From our limited point of view, percolation is just another way of producing random graphs with plenty of known analytical results. The process resembles greatly that of the previous section; the main difference is that now in general, the vertex set is not limited to a fixed number and edges may appear only between certain vertices instead of all possible positions.

[1] A property $\mathcal{P}$ is *convex* if $F \subset G \subset H$, F and H having the property $\mathcal{P}$ implies that G also has $\mathcal{P}$ [19].

Consider the infinite square lattice $\mathbb{Z}^2$ with a possibility of placing a vertex at each "crossing" of the grid and an edge at each unit grid line. An infinite random graph $G = (V, E)$ is formed by taking the set of all crossings as vertices and selecting the edge set by including each unit grid line randomly and independently with probability $p \in [0, 1]$. A smaller graph results if we restrict ourselves to a specified portion of $\mathbb{Z}^2$, usually some rectangular area. The central question in *percolation theory* is the following: In a random graph constructed on a finite portion of $\mathbb{Z}^2$, does an *open path* exist from one boundary of the rectangle to the opposite boundary? An example of a rectangular graph that contains such a path is shown in Figure 3.1.

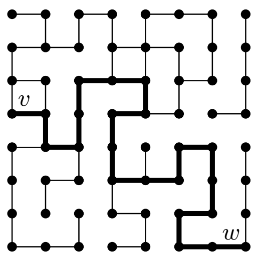


FIGURE 3.1: A grid with randomly placed edges that has an open path from v to w.

As p is varied, the structure of the random graph begins to change. After reaching some *critical probability* p_c, the infinite graph contains (with probability one) an infinite *cluster* of connected vertices. Such a procedure of adding edges is called *bond percolation* and it could of course be conducted on other structures besides the square lattice $\mathbb{Z}^2$. Another, somewhat less studied variant is *site percolation*, where instead of adjusting the edge presence, the existence of a vertex is decided upon independently and randomly with probability p. The edge set consists of one-unit grid lines that connect two vertices that are both present.

Percolation phenomena have been widely studied in physics, for example as models of magnetism. Also studies of epidemic spreading resort to percolation as a mathematical model: Newman and Watts [106] have studied site percolation in so-called *small-world networks*. These results will be summarized in Section 4.1; the construction of such networks is presented in Section 3.2.1.

3.1.3 Generating graphs to match a degree distribution

A disadvantage of the uniform models is that the degree distributions in natural networks, such as the power-law distribution observed for the Internet and the Web graph, differ significantly from the Poisson distribution of random graphs. This suggests that the ER model is not adequate for natural networks, as important features of the underlying phenomena are being ignored [12, 105]. It was already suggested by Erdős and Rényi in [44] that in "a real situation, one should replace the hypothesis of equiprobability of all connections by some more realistic hypothesis". To better match the structure of natural networks, random networks that follow some predetermined degree distribution are of interest. This section describes two approaches for generating such graphs.

Molloy and Reed [96] study the *degree sequences* $\boldsymbol{d} = (d_1, \ldots, d_n)$ of graphs; these are unique when sorted in either increasing or decreasing order. Each degree sequence characterizes a family of graphs. A graph $G = (V, E)$ belongs to the ensemble of degree sequence $\boldsymbol{d}$, is the vertices of G can be labeled so that d_i is the degree of v_i for all $i = 1, 2, \ldots, n$. A graph with a specified degree sequence is a uniform random sample drawn from the corresponding family. As properties are defined for random graphs with a specified degree sequence, the calculations are in fact averages over such families. A *degree distribution* may also be defined as a list of probabilities $\boldsymbol{p} = (p_1, \ldots, p_n)$ where $p_k = \Pr[\deg(v) = k]$ for an arbitrary vertex $v \in V$.

Molloy and Reed construct a graph $G = (V, E)$ to match a given degree distribution by the following algorithm, interpreted as in [107]: attach to vertex v_i exactly k_i "stubs" (half-edges that lack the other end-point), where k_i is drawn randomly and independently from $\boldsymbol{p}$. If it happens that there is an odd number of stubs in total, replace the stubs of a randomly chosen vertex with a new set of k stubs, k again randomly drawn, and repeat this until an even number of stubs are present in total. After successfully assigning stubs to all vertices, choose two random stubs and merge the selected stubs into a proper edge. We note that multiple and reflexive edges can be easily avoided in this step if desired. Merging of random stubs is continued until no stubs remain. Many properties of this model have been derived by Newman, Strogatz and Watts [105, 107], such as the average distance, which was found to be logarithmic in n. In [105], the authors also consider directed graphs.

Mihail et al. [94] construct a graph $G = (V, E)$, $V = \{v_1, v_2, \ldots, v_n\}$ to match a given degree distribution $\boldsymbol{d}$ using Markov chains. The degree sequence $\boldsymbol{d} = (d_1, d_2, \ldots, d_n)$, sorted in decreasing order of the degrees such that $d_1 \geq d_2 \geq d_3 \geq \ldots \geq d_n)$ is said to be *realizable* if a graph that matches $\boldsymbol{d}$ exists. It has been shown (see [94] and the references therein) that the following condition is both necessary and sufficient for a degree sequence $\boldsymbol{d}$ to be realizable:

$$\sum_{i=1}^{k} d_i \leq k(k-1) + \sum_{i=k+1}^{n} \min\{k,\, d_i\}, \quad \text{where } 1 \leq k \leq n-1. \tag{3.4}$$

The algorithm maintains a list of *residual degrees* of the vertices, $\deg_r(v_i) = d_i - \deg(v_i)$. In each iteration of the construction algorithm, a vertex v_i is picked and connected to $\deg_r(v_i)$ vertices of *highest* residual degree, updating the residual degree of v_i to zero and reducing the residual degrees of the endpoints of the edges by one. Proceeding in this manner ensures that the condition of Equation 3.4

stays valid after the iteration for those values of $\boldsymbol{d}$ that are yet to be satisfied. This process runs in linear time and may also be modified to link the vertices with preferential attachment. If the algorithm constructs a disconnected graph, it must contain a cycle if $\sum_i d_i \geq 2(n-1)$. Removing an edge (u, v) from the cycle and an edge (s, t) from a component that is not connected to the cycle, and replacing them by the edges (u, s) and (v, t) does not affect $\boldsymbol{d}$. As u is now connected to s and v to t, the two components have become connected. This may be repeated as long as there are multiple components.

An interesting question is how to obtain a *random* instance from the ensemble $\mathcal{G}(\boldsymbol{d})$ of graphs that share the same degree sequence $\boldsymbol{d}$. Mihail et al. [94] achieve this by running a *Markov chain*[2] starting with any realization $G \in \mathcal{G}(\boldsymbol{d})$ obtained by the above algorithm. The process picks two edges (u, v) and (s, t) at random from G, ensuring that the endpoints are all distinct, and replaces these by the edges (u, s) and (v, t) to obtain G'. If G' is disconnected, the switching operation is canceled. From the theory of Markov chains (see [94] and the references therein), it is known that performing such perturbations will reach every possible graph in $\mathcal{G}(\boldsymbol{d})$ with equal probability in the limit, independently of the start position $G \in \mathcal{G}(\boldsymbol{d})$.

Mihail et al. [94] recommend using the following stop condition to detect when the graph has become sufficiently random: keep sorted adjacency lists for all vertices that have a unique degree in the starting topology G, compute such lists also for the current topology G' and count the number of positions in which these lists differ. The larger the count, the more "different" the graphs G and G' are expected to be. In their simulations, Mihail et al. have found that this measure first increases linearly before leveling off at some time T. For graphs of order 12,000 or less they recommend running for $3T$ time steps. To estimate the exact mixing speed, the authors studied a graph of order 11,000 for which they found $T < 180{,}000$. The graphs generated by this algorithm are static and bound to one degree sequence. Mihail et al. propose their model for generating Internet-like graphs; a setback is that the Internet grows and thereby changes its degree sequence frequently.

3.2 Small-world networks

When attending a cocktail party, people frequently notice to their surprise that although they are not previously acquainted with each other, they share common acquaintances. In the 1960s, Stanley Milgram studied this phenomenon by estimating the number acquaintances needed to pass a letter hand to hand, coast to coast in the United States. Although the outcome of his experiment was somewhat

[2] A Markov chain is a stochastic process $\{X\}_{n\geq 0}$ that satisfies the following "memoryless" condition: for all $n \geq 1$ and all x_i in the state space, $\Pr[X_n = x_n \mid X_1 = x_1, \ldots, X_{n-1} = x_{n-1}] = \Pr[X_n = x_n \mid X_{n-1} = x_{n-1}]$ [62].

entangled (see e.g. [81]), he concluded that the length of the necessary acquaintance chain is usually six. This was such a catchy result that it became a common phrase and even resulted in a play by John Guare titled "*Six Degrees of Separation*" [63]. The six-degrees phenomenon has been also known as the small-world phenomenon due to the title of Milgram's article, "The small world problem" [95]. It would indeed be a small world if all U.S. citizens would be just a few acquaintances away from each other. Another observation of social sciences is that in a network of friends, two people who are both friends with a third person are more likely to be friends with each other than with a randomly chosen person [104]. The tight community of friends of a certain individual is often called the *cluster* of that individual.

In this section we discuss models that aim to generate networks that resemble the social networks in these two aspects: the average distance from one node to another is small, and the network consists of dense clusters. In uniformly generated random graphs, the distance between any two vertices is rather small, but there is no tendency toward cluster formation, as all edges are equally likely. Altering the degree distribution does not produce the desired clustering either, as the high-degree vertices may easily be placed far apart instead of linking them into a dense neighborhood. We now describe some models proposed to cope with this difficulty. For an extensive review of the work leading to this genre of network modeling, see either Barabási's book "*Linked*" [11] on complex networks in general, or Watts's book "*Small Worlds*" [133], which is more mathematical than Barabási's popularization of the topic.

3.2.1 The Watts-Strogatz model: Random rewiring

In 1998, Watts and Strogatz [134] brought the small-world phenomenon to the attention of researchers in various fields. They presented a simple procedure for randomizing networks of a certain structure and argued that the resulting networks have a property often seen in natural networks that resembles the small-world phenomenon of social networks. They generate *small-world networks* with the following procedure, interpreted as in [16], which we denote as the WS model. The initial graph is a *circulant graph* $C_{n,k}$ of n vertices where each vertex is connected to $2k$ of its *nearest neighbors*, k on each direction along the ring.[3] An example of a circulant graph is given on the left in Figure 3.2.

Watts and Strogatz introduce randomness into the initial graph by selecting a vertex v and the edge (v, w) connecting it to the next vertex w on the ring. With probability p, the edge (v, w) is *rewired* by replacing w with a random vertex $u \in V$,

[3]It appears to be a matter of definition whether the notation $C_{n,k}$ is used to indicate k neighbors on *each side* of a vertex or k neighbors in total, $k/2$ on each side. The latter would require k to be even, which may be at times confusing. We therefore adopt the former convention.

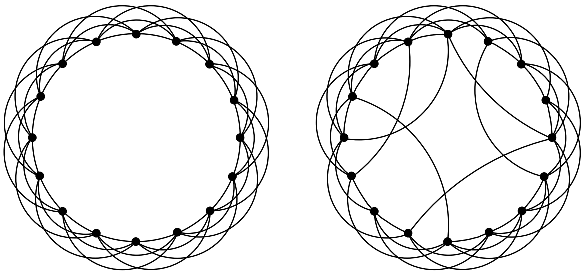


FIGURE 3.2: On the left, the circulant graph $C_{16,3}$; a randomly rewired version (with small p) on the right (adapted from [134]).

with multiple edges forbidden. This is repeated for each vertex along the ring. Then a second round of rewiring takes place, now concerning the edges connecting "second-neighbors" on the ring, completing in total k rounds of rewiring. The above procedure may produce disconnected graphs, which poses challenges to the analysis as well as the use of many common metrics. Therefore Watts and Strogatz require that $1 \ll \ln n \ll 2k \ll n$, where $2k \gg \ln n$ guarantees that the resulting random graph is connected [19].

Newman et al. [104, 106] modify the WS model to avoid disconnected instances. Instead of rewiring edges of the circulant graph $C_{n,k}$, they add some random edges. The shortcuts are created by randomly selecting nk vertex pairs $\{u, v\}$, one for each edge of the underlying $C_{n,k}$, and adding the edge (u, v) with probability ϕ, generating on average $nk\phi$ shortcuts.[4] This version of the model, the SWS model, appears to be the established version of the WS model; our implementation is described in Section 5.1.2 with some properties of the model. Some small-world models allow other underlying graphs than $C_{n,k}$; one such model is briefly discussed in Section 3.2.4.

The two properties of the WS model examined by Watts and Strogatz are a global property, the *characteristic path length* $\mathcal{L}(p)$ of the graph, and a local property, the *clustering coefficient* $\mathcal{C}(p)$. These are defined loosely as follows: $\mathcal{L}(p)$ is a measure of "typical separation between two vertices in a graph" and $\mathcal{C}(p)$ is a measure of "cliquishness of a typical neighborhood". These two measures are used in [134] and many other publications to determine whether a given network has "the small-world property". Watts and Strogatz loosely define that if for a given graph G, $\mathcal{C}(G)$ is relatively high and $\mathcal{L}(G)$ is as small as a random graph of the same order and size would have, then G has the small-world property.

Definition 3.1. The *characteristic path length* $\mathcal{L}(G)$ of a graph $G = (V, E)$ is the average length of the shortest path between two vertices in G.

[4]It appears that u and v are implicitly distinct and previously nonadjacent vertices.

Definition 3.2. The *clustering coefficient* $\mathcal{C}(G)$ of a graph $G = (V, E)$ is the average clustering coefficient of its vertices $v \in V$. The clustering coefficient $\mathcal{C}_v \in [0, 1]$ of a vertex v is the density of the subgraph induced by $\Gamma(v)$.

Note that $\mathcal{C}_v = 1$ if $\Gamma(v)$ forms a clique. Therefore $\mathcal{C}(K_n) = 1$. Note also that the local clustering coefficient of a vertex that has only one neighbor is a matter of definition as the divisor is zero and the result therefore undefined. We exclude such vertices in our calculations when averaging to obtain $\mathcal{C}(G)$ for a particular graph G. For a $\mathcal{G}_{n,p}$ graph, obviously $\mathrm{E}[\mathcal{C}] = p$, as the $\mathrm{E}[\delta] = p$ for any subgraph, including the neighborhood graphs. Another noteworthy point is that $\mathcal{C}(v)$ is a measure for the relative number of triangles [65, 107]:

$$\mathcal{C} = \frac{3 \times \text{the number of } K_3 \text{ subgraphs}}{\text{the number of } P_3 \text{ subgraphs}}. \tag{3.5}$$

It is also noteworthy that although the clustering coefficient is locally the same as density, these measures are not globally dependent on each other. Consider a graph G that consists of two K_h subgraphs that are connected to each other by just one edge. There are $2h$ vertices in the graph and $h(h-1)+1$ edges. Therefore the density is

$$\delta = \frac{h^2 - h + 1}{2h^2 - h}, \tag{3.6}$$

which approaches $\frac{1}{2}$ as h grows larger. All other vertices but the ones adjacent to the "bridge" between the two cliques have $\mathcal{C} = 1$, the two others having $\mathcal{C} = 1 - \frac{2}{h}$. Hence $\mathcal{C}(G) = 1 - \frac{2}{h^2}$, which approaches one as h grows. On the other hand, a complete bipartite graph $K_{h,h}$ has density $\delta = h/(2h-1)$, which is also close to $\frac{1}{2}$ for large h, but has by definition $\mathcal{C} = 0$, as the neighborhood of any vertex is an independent set. Graphs of similar density may therefore have entirely different values of $\mathcal{C}$. Toby Walsh [131] proposes the following "quantitative measure" for the presence of the small-world phenomenon:

Definition 3.3. The *proximity ratio* μ of a graph G is the ratio between the clustering coefficient $\mathcal{C}$ and characteristic path length $\mathcal{L}$ of G normalized by the same measures for a random graph of the same order and size, $\mathcal{C}_r$ and $\mathcal{L}_r$:

$$\mu = \frac{\mathcal{C}\mathcal{L}_r}{\mathcal{C}_r\mathcal{L}}. \tag{3.7}$$

Watts and Strogatz [134] compare the characteristic path length and clustering coefficient of a WS graph to the respective values for the circulant graph. To analyze $\mathcal{L}(C_{n,k})$ and $\mathcal{C}(C_{n,k})$ we first derive some properties of the circulant graphs. In a $C_{n,k}$, the degree of each vertex $v \in V$ is exactly $2k$ and hence it is $2k$-regular. The number of edges in $C_{n,k}$ is $m = nk$, as each vertex has $2k$ neighbors and summing

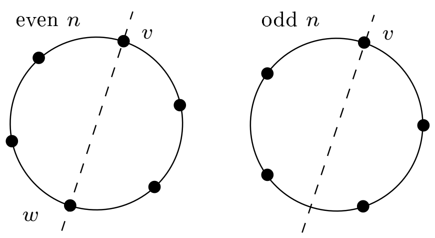


FIGURE 3.3: Two $C_{n,1}$ graphs split in two from vertex v.

over the n vertices counts each edge twice. From this we obtain $\delta(C_{n,k}) = \frac{2k}{n-1}$. We also point out that when $k \geq \lfloor \frac{n}{2} \rfloor$, $C_{n,k} \cong K_n$ and therefore has diameter one.

To derive a generic formula for the diameter, fix a vertex $v \in V$ and split the "ring" in two from v as shown in Figure 3.3. For even values of n, there will be a vertex directly opposite to v which we denote by w. The distance from v to w is necessarily the diameter of the graph. Between v and the possible opposite vertex w, there are r vertices on both sides,

$$r = \begin{cases} \frac{n-2}{2}, & n \text{ even}, \\ \frac{n-1}{2}, & n \text{ odd}. \end{cases} \tag{3.8}$$

The first k of the r vertices in either direction along the ring are at distance one from v, the next k at distance two, and so forth. To determine the diameter of $C_{n,k}$, we observe that each "step" may take us as far as k vertices forward. Hence

$$\operatorname{diam}(C_{n,k}) = \begin{cases} \lceil \frac{n}{2k} \rceil, & n \text{ even}, \\ \lceil \frac{n-1}{2k} \rceil, & n \text{ odd}. \end{cases} \tag{3.9}$$

The number of "full blocks" of k vertices on each side of the ring between v and the opposite position (which is empty for odd n and the vertex w for even n) is

$$b = \begin{cases} \lfloor \frac{n-2}{2k} \rfloor, & n \text{ even}, \\ \lfloor \frac{n-1}{2k} \rfloor, & n \text{ odd}. \end{cases} \tag{3.10}$$

We are now ready to define formula for $\mathcal{L}$ of $C_{n,k}$. The sum of distances from v to the b full blocks of vertices on both sides of the ring is twice the sum of distances to one side:

$$2 \sum_{i=1}^{b} ki = kb(b+1). \tag{3.11}$$

In addition, there are $r \pmod{k}$ vertices at distance $b+1$ on each side. We add the distance to these vertices to Equation 3.11 to obtain the total distance from

v to all other vertices in the graph excluding the possible vertex w as $D = (b+1)(2r-kb)$. The sum of all distances within $C_{n,k}$ is $T = nD$ for odd n and $T = n(D+\mathrm{diam}(C_{n,k}))$ for even n. As there are $n(n-1)$ distances included in this sum, the average distance $\mathcal{L} = \frac{T}{n(n-1)}$. Making some substitutions and simplifying, we obtain

$$\mathcal{L}(C_{n,k}) = \begin{cases} \frac{1}{n-1}\big((\lfloor\frac{n-2}{2k}\rfloor+1)(n-2-k\lfloor\frac{n-2}{2k}\rfloor)+\lceil\frac{n}{2k}\rceil\big), & n \text{ even}, \\ \big(\lfloor\frac{n-1}{2k}\rfloor+1\big)\left(1-\dfrac{k\lfloor\frac{n-1}{2k}\rfloor}{n-1}\right), & n \text{ odd}. \end{cases} \tag{3.12}$$

Asymptotically for large n and fixed k, this yields $\mathcal{L} \sim \frac{n}{4k}$, in accordance with the result of Watts and Strogatz [134] that for the WS model $\mathcal{L} \sim \frac{n}{4k} \gg 1$ when $p \to 1$. This estimate can be obtained by approximating the diameter of the graph by $\frac{n}{2k}$, assuming the distances to take uniformly values from one to the diameter, thereby obtaining an average of about half the diameter; hence $\mathcal{L} \sim \frac{n}{4k}$. Watts and Strogatz also state that $\mathcal{C} \sim \frac{3}{4}$ when $p \to 0$, for which we show a derivation due to Comellas et al.:

Theorem 3.1. [29] The clustering coefficient of $C_{n,k}$ is $\mathcal{C}(C_{n,k}) = \dfrac{3(k-1)}{2(2k-1)}$.

Proof. All vertices $v \in V$ have same the clustering coefficient $\mathcal{C}_v$ for $C_{n,k}$ due to the symmetric structure. A given vertex v_i has $2k-2$ common neighbors with its immediate neighbor v_{i+1} (or symmetrically v_{i-1}) on the ring.

Therefore $|\Gamma(v_i) \cap \Gamma(v_{i\pm1})| = 2k-2$. Similarly $|\Gamma(v_i) \cap \Gamma(v_{i\pm j})| = 2k-(j+1)$, $1 \le j \le k$. Summing over the neighbors of v_i gives the number of triangles $\mathcal{T}_{v_i}$ that contain v_i:

$$\mathcal{T}_{v_i} = \sum_{j=1}^{k} \big(2k-(j+1)\big) = \frac{3k(k-1)}{2}. \tag{3.13}$$

Vertex v_i has k neighbors on each side, which introduces a factor 2, which cancels out as each triangle is counted twice in the sum. Note that $\mathcal{T}$ is the same for all vertices due to the symmetry of the graph. As $\mathcal{T}_v$ is equal to the number of edges present in $\Gamma(v)$,

$$\mathcal{C}(C_{n,k}) = \frac{3k(k-1)}{2\binom{2k}{2}} = \frac{3(k-1)}{2(2k-1)}. \tag{3.14}$$

$\square$

We now return to discuss the small-world property. For $\mathcal{G}_{n,m}$ with $m = nk$, Watts and Strogatz [134] have derived $\mathcal{L}_r \sim \frac{\ln n}{\ln k}$ and $\mathcal{C}_r \sim \frac{2k}{n} \ll 1$. For random graphs, $\mathcal{L}_r \sim \frac{\ln n}{\ln \overline{k}}$ (see e.g. [26] for this and more general results), where $\overline{k} = p(n-1)$ for

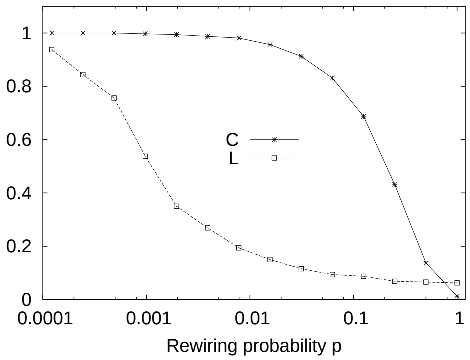


FIGURE 3.4: Values of $\mathcal{C}(G)$ and $\mathcal{L}(G)$ normalized by $\mathcal{C}(C_{n,k})$ and $\mathcal{L}(C_{n,k})$ respectively for $p \in (0, 1]$. Sample graphs are generated for parameter values $p = 1, 0.5, 0.25, 0.125, \ldots$ until $p < 0.0001$. The values are averages over 20 random realizations of the WS model for $n = 1,000$ and $k = 5$. Data is interpreted from a figure in [134].

$\mathcal{G}_{n,p}$ and $\overline{k} = \frac{2m}{n}$ for $\mathcal{G}_{n,m}$. Now that $m = nk$, $\mathcal{L} \sim \frac{\ln n}{\ln k}$. The density of a $G_{n,nk}$ is $\delta = 2k/(n-1)$, which for large n is close to $2k/n$. For a uniform random graph $\mathcal{C}$ should be similar to the density and therefore it is justified to state $\mathcal{C} \sim \frac{2k}{n}$. Note that for $n \gg k$, $\mathcal{C} \ll 1$.

As the rewiring probability goes to one, $\mathcal{L} \approx \mathcal{L}_r$ and $\mathcal{C} \approx \mathcal{C}_r$ even though the process does not produce truly random graphs. The problem is that only one endpoint of the rewired edge is "randomized"; the source of the edge will not change. As each edge from a vertex v to its k neighbors in one direction is rewired once, the degree of v will remain unaffected. It may lose neighbors as the k vertices preceding v are rewiring their edges, but it will certainly maintain a degree at least k. This would not be true for a $G_{n,nk}$, and therefore the rewiring procedure does not generate true random graphs. However, much of the symmetry of the original graph is lost in the rewiring, which justifies to some degree the classification of the rewired graphs as "random".

Figure 3.4, adapted from [134], shows the "small-world" effect in more concrete terms: $\mathcal{C}$ remains practically unaffected as $\mathcal{L}$ drastically drops after very little rewiring — note that in Figure 3.4 the x-axis has logarithmic scale.[5] Watts and Strogatz [134] define *small-world networks* to be those networks for which $\mathcal{L}(p)$ is nearly as small as $\mathcal{L}_r$, whereas $\mathcal{C}(p)$ is significantly larger than in random networks, $\mathcal{C}(p) \gg \mathcal{C}_r$. Note that the rewiring probability required for the small-world phenomenon to appear is very small, $p \approx 0.01$. Watts and Strogatz also inspect the effect of the small-world property in a dynamical system: a simplified model

[5]On page 87, Figure 5.1 displays the same curves for our implementation of the SWS model (a WS variant presented later in this section) and additionally the unscaled curves together with those for random networks of the same order and similar size.

TABLE 3.1: Examples from previously encountered natural network models with the values of the small-world measures listed. Note that $\mathcal{C}_r$ is practically the density δ of both the corresponding $G_{n,m}$ and the network in question. The calculations on the first row are our own.

Network	$\|V\|$	$\|E\|$	$\mathcal{L}$	$\mathcal{L}_r$	$\mathcal{C}$	$\mathcal{C}_r$	μ	Ref.
C. elegans	202	1,954	2.28	2.04	0.30	0.096	2.9	
C. elegans	282	1,974	2.65	2.25	0.28	0.05	4.8	[134]
`.edu`	11,000		4.06	4.05	0.16	0.0012	129.6	[3]
IMDb	225,226	6,869,393	3.65	2.99	0.79	0.00027	2396.9	[134]
Internet AS	12,709	27,384	3.62	3.64	0.46	0.0014	330.2	[24]
Power grid	4,941	6,596	18.7	12.4	0.08	0.005	10.6	[134]
Thesaurus	30,244		3.16	2.5	0.53	0.002	209.7	[98]

of epidemic spreading[6], predicting that infectious diseases spread more rapidly in small-world networks. They also touch upon a few other possibly interesting research topics.

Watts and Strogatz suspected that the "small-world phenomenon might be common in sparse networks with many vertices, as even a tiny fraction of shortcuts would suffice" [134]. They examined their conjecture on the largest connected components of some natural networks; values of $\mathcal{L}$, $\mathcal{C}$, and μ of these and other natural networks are shown in Table 3.1. In the sense of small $\mathcal{L}$ together with a large $\mathcal{C}$, all of the models in Table 3.1 display the small-world phenomenon. Note that the magnitude of μ can be very different for networks that are all small worlds as defined by Watts and Strogatz [134]. The models listed in Table 3.1 are quite simple. For example, the data concerning the World Wide Web from [3] concerns a graph drawn with the sites visited by a certain webcrawl. The paper also provides measures for the entire crawl and the largest connected component. We omit these here, as the calculations are only estimates due to the large network size; the subgraph of `.edu`-domains is sufficiently small to calculate the exact values for these measures.

Values of $\mathcal{C}_r$ and $\mathcal{L}_r$ are averages of over 30 randomly generated $\mathcal{G}_{n,m}$ graphs. If only n and $\overline{k}$ were given in the reference, we used the equality $m = \frac{n\overline{k}}{2}$ rounding to an integer. Some of the cited articles readily provide values of $\mathcal{C}_r$ and $\mathcal{L}_r$, but these may be calculated for just one $G_{n,m}$ and are not necessarily characteristic. We list these values instead of calculating our own only if n and m or $\overline{k}$ are not reported in the referenced publication. In some articles it is not clear whether the values of $\mathcal{L}$ and $\mathcal{C}$ are only for the largest connected component, and whether they obey edge-direction for directed models.

These problems reflect the general difficulty of employing such network measures. Surprisingly, also general observations on the behavior of these measures have been

[6]Modeling epidemic spreading with networks is briefly addressed in Section 4.1.

made. Vázquez, Pastor-Satorras, and Vespignani [129] report several power-laws and other relations, including that of the Internet: $\mathcal{C}_k \sim k^{-\omega}$, for a vertex with degree k, where $\omega = 0.75 \pm 0.03$. Such laws are helpful in situations where an exact calculation is tedious.

3.2.2 Kleinberg's lattice model

Jon Kleinberg argues that the WS model does not succeed in capturing the algorithmic aspect of Milgram's original research; if letters do propagate efficiently from coast to coast, it certainly suggests that "individuals using local information are collectively very effective at actually *constructing* short paths between two points in a social network" [79]. Kleinberg shows that for the WS model, there cannot exist a decentralized algorithm operating only on local information that could construct such short paths. He suggests a modification of the WS model to capture this behavior. For graphs generated by the Kleinberg lattice model (KL), there exists a decentralized algorithm that will find the desired short paths with high probability. Instead of the circulant graph, the ambient network in the Kleinberg model is an $s \times s$ grid in which vertices are pairs $v = (i, j)$, $i, j \in \{1, \ldots, s\}$. The radius within which local edges are present is fixed to $p \geq 1$, using the *Manhattan distance*

$$dist_{\mathbb{L}}(u, v) = dist_{\mathbb{L}}((i, j), (k, \ell)) = |k - i| + |\ell - j|. \tag{3.15}$$

For $p = 1$, a grid appears (see Figure 3.5). In addition to these local connections, a fixed number $q \geq 0$ of directed long-range edges are assigned to each vertex $v \in V$ randomly and independently: $\langle v, w \rangle$ is chosen with probability proportional to $d\,(v, w)^{-r}$, where $r \geq 0$ is a constant. No duplicate edges are allowed, which also excludes the vertices within Manhattan distance p of v when selecting its q additional neighbors.

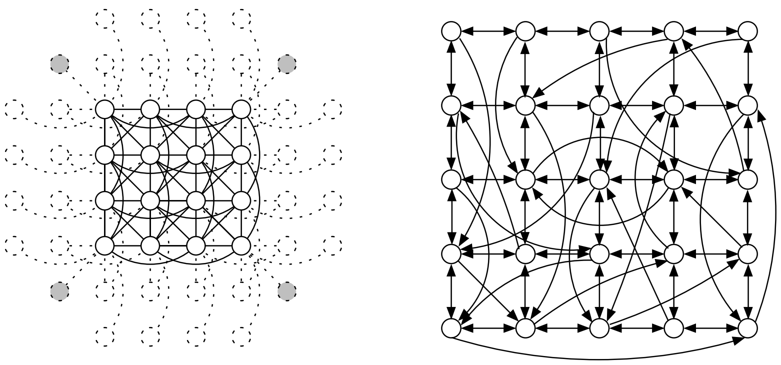


FIGURE 3.5: On the left, a KL graph with $s = 4$, $p = 2$, and $q = 0$. On the right, a KL graph with $s = 5$ and $p = q = 1$.

Clearly the order of the graph is $n = s^2$ and it is connected for $p \geq 1$. The number of edges is less obvious. The order of a p-neighborhood that is completely included in the lattice is

$$\deg_p(v) = |\{u \mid dist_{\mathbb{L}}(u,v) \leq p\}| = \sum_{i=1}^{p-1} 4i = 2p(p-1), \tag{3.16}$$

and hence $2m < n \deg_p(v)$. This upper bound contains "ghost" p-neighbors of vertices near the border of the lattice that are not included in the graph (drawn in Figure 3.5 with dotted line). There are s vertices on each of the four sides of the lattice. A vertex at distance $d \in [0,p)$ from the border has $\sum_{j=1}^{p-d}(2j-1)$ ghost neighbors, which gives a total of

$$4s \sum_{i=0}^{p-1} \sum_{j=1}^{p-i} (2j-1) = 4s\left(p^3 - p^2(p-1) - \frac{p(p-1)(2p-1)}{6}\right). \tag{3.17}$$

In this count, some of the ghost neighbors in the four corner areas (shaded in the example graph of Figure 3.5) are included twice. The number of such vertices is

$$\sum_{\substack{i,j<p \\ i+j\leq p}} \sum_{k=1}^{p-(i+j)-1} k. \tag{3.18}$$

By the Inclusion-Exclusion Principle, the number of undirected neighbors is obtained by subtracting the ghost neighbors from the upper bound and adding back those that were doubly subtracted. Dividing this by two to obtain the number of undirected edges and adding the qn directed edges, we obtain the size of the graph m. Our undirected implementation of this model is presented in Section 5.1.3.

In experimental studies of his network model, Kleinberg [79] concluded that $r = 2$ is the only integer value for which any decentralized algorithm is able to reach any vertex from any other vertex by traversing a path of length $\mathcal{O}(\log n)$ using *only local information* on the network structure. Note that when $r = 0$, the probability of an edge being present will no longer depend on the separation distance, and hence the distribution of long-distance edges is uniform. For $r \neq 2$, Kleinberg states that the expected "delivery time" (for Milgram's letters) of any decentralized algorithm is higher. Based on this observation of a unique value $r = 2$, Kleinberg argues that not all small-world networks are in general quickly navigable. He generalizes this result to d-dimensional lattices, where the optimal value is $r = d$ and suggests that "the correlation between local structure and long-range connections provides critical clues for finding paths through the network", not only in the lattice model, but small-world networks in general [78].

3.2.3 Connected caveman graphs

An early attempt in social sciences to capture the clustering properties of social networks was the *caveman graph*, produced by linking together a ring of small complete graphs called "caves" by moving one of the edges in each cave to point to another cave (see [133] and the references therein). Figure 3.6 illustrates this principle. Note that the individuals of one cave will be connected closely to each other and the populations of the separate caves are connected sparsely.

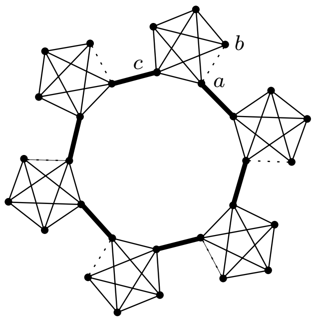


FIGURE 3.6: A ring of six K_5 caves rewired to form a connected graph. The rewired edges are shown with a dotted line, the replacements are drawn thicker (adapted from [133]).

Watts [133] derives the clustering and path length properties for a caveman graph $G = (V, E)$, $|V| = h(k+1)$, that consists of h "caves" isomorphic to K_{k+1}. Clearly $|E| = h\binom{k+1}{2}$ and hence $\delta = \frac{k}{h(k+1)-1}$. The clustering coefficient of one cave is equal to that of the entire graph. The vertices in each cave may be classified into the following four types (see Figure 3.6):

(a) one vertex v_a with $\deg(v_a) = k$ from which an edge was rewired to the next cave; the new neighbor is not connected to any of the k others that are all mutual neighbors — hence $\mathcal{C} = \frac{\binom{k}{2}-(k-1)}{\binom{k}{2}} = 1 - \frac{2}{k}$,

(b) one vertex v_b with $\deg(v_b) = k - 1$ that lost the rewired edge; all of the remaining neighbors are connected and hence $\mathcal{C}(v_b) = 1$,

(c) one vertex v_c with $\deg(v_c) = k + 1$ that gained a new neighbor from the rewired edge that is not connected to the k other neighbors; within the old neighbors one rewired edge is missing and hence $\mathcal{C}(v_c) = \frac{\binom{k+1}{2}-(k+1)}{\binom{k+1}{2}} = 1 - \frac{2}{k}$.

(d) $k - 2$ other vertices v_d with $\deg(v_d) = k$, for which the only edge missing from the K_{k+1} is the rewired edge; hence $\mathcal{C}(v_d) = \frac{\binom{k}{2}-1}{\binom{k}{2}} = 1 - \frac{2}{k(k-1)}$.

Taking a weighted average of the above we obtain

$$\mathcal{C}(G) = \frac{1}{k+1}\left(k + 1 - \frac{6k-8}{k(k^2-1)}\right). \tag{3.19}$$

As the caves are grown larger, the fraction of vertices with high clustering grows and eventually $\mathcal{C} \to 1$. Watts [133] also calculates the characteristic path length, obtaining

$$\mathcal{L} = \frac{8}{k(k+1)} + \frac{\left(\frac{n}{k+1}\right)^2}{2\left(\frac{n}{k+1} - 1\right)} + 1, \tag{3.20}$$

based on reasoning about the average distances inside a cave and the distances required to move from one cave to another. Intuitively, the asymptotic average distance is half the diameter, which is approximately $\operatorname{diam}(C_{h,1})$ for large graphs, as the caves are connected as vertices of $C_{h,1}$ and moving within a cave takes at most one extra step.

In our experiments, reported in Chapter 5, we consider a hierarchical caveman graph construction to model coauthorship networks. One cave is considered to represent a research group, the members of the group being closely connected. Several research groups are linked to form a laboratory, and furthermore several laboratories are loosely connected to form a department, etc.

3.2.4 Alternative models and measures of small-world networks

The above definitions and models of the small-world property have not been entirely satisfactory, leading to other approaches. The question of the most accurate model is still to be settled and variations come up frequently; therefore this text does not attempt to be a full review of the field, but rather a glance at some of the recent suggestions. Pandit and Amritkar [111] define "shortcuts" in graphs that are not tied to a certain graph topology. They call such shortcuts the *far edges* of the network.

Simply stating, (i, j) is a *far edge* of order μ if no simple path of length $\mu + 1$ exists from i to j (for a formalization, see [111]). The minimal order of (i, j) is $\mu_{\min}$ if there is at least one path of length $\mu_{\min}$ connecting i to j but not a path of length $\mu_{\min} + 1$. A far edge with $\mu_{\min} = 1$ is hence defined as an edge that is not included in any triangle [56]. Note that no edge in K_n can be a far edge and all edges of a tree are far edges of all orders.

Denoting the ratio of far edges to $|E|$ by $\mathcal{F}$, Pandit and Amritkar [111] find by experiments that for the small-world region of the WS-graphs, where $\mathcal{C}(G) \approx \mathcal{C}(C_{n,k})$ and $\mathcal{L}(G) \approx \mathcal{L}(G_{n,p})$, this ratio is small: $\mathcal{F} \approx 0.01$. They suggest that $\mathcal{F}$ would be a better parameter for small-world generation than p, as it is not in any way dependent on the generation method, whereas the p of the WS model is strictly limited to the case of rewiring a $C_{n,k}$.

Gent, Hoos, Prosser and Walsh [53] obtain networks with small-world properties by a mechanism they call *morphing*. They start with two structures (e.g. graphs) and combine (parts of) these to produce a structure that has properties of both the original structures. For graphs, the procedure is the following: take two graphs with a common vertex set V of size n, denote these as $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$. Form the "morph" $G_m = (V, E_m)$ by including $E_1 \cap E_2$ in E_m and taking in addition a fraction p of edges in $E_1 \setminus (E_1 \cap E_2)$ and a fraction $1-p$ of those in $E_2 \setminus (E_1 \cap E_2)$.

Gent et al. also define a *matrix morph* in which two $m \times n$ matrices A_1 and A_2 are combined into A_m by choosing each of the elements randomly and independently from the two possible choices, from A_1 with probability p and from A_2 with probability $1 - p$. This too can be used to produce a graph, as graphs are essentially captured by their adjacency matrices. Also incidence matrices[7] of graphs can be used. Also other kinds of morphs are defined in Gent et al. [53], but they are of little interest here.

To produce a small-world network from two easily obtainable graphs, the initial graphs G_1 and G_2 should be chosen so that the other has high clustering but high average path length (e.g. $C_{n,k}$) and the other has small diameter (e.g. $G_{n,p}$). The advantage of this method over the WS model is, according to Gent et al. [53], that "the theoretical analysis of morphs is likely to be much easier than that of rewired graphs". They have experimented with morphs of the graph type described above and found that the behavior of $\mathcal{L}$ and $\mathcal{C}$ is very similar in morphing as it is in the WS model. They also compare the behavior of Walsh's proximity ratio μ (see Definition 3.3 on page 32), again finding curves of similar shape.

Latora and Marchiori [86, 90] in turn criticize the original WS model for limited scope and propose a generalization to weighted graphs (allowing further generalization to disconnected or dense graphs). Instead of speaking in terms of the clustering coefficient $\mathcal{C}$ and characteristic path length $\mathcal{L}$, they define the small-world phenomenon first in terms of *connectivity length* $\mathcal{D}$ in [90] and later in terms of *efficiency* $\mathcal{E}$ in [86].

The connectivity length $\mathcal{D}$ is introduced to embrace physical distances into determining the presence of the small-world property. It is a valid measure for any metrical graph and portrays the efficiency of information propagation defined by the *separation distances* $d_s(u, v)$, $u, v \in V$. A separation distance is defined to be the smallest sum of physical distances over the set of paths connecting u to v in G.

[7] The rows of an $n \times m$ incidence matrix $\mathbf{M}$ of a graph $G = (V, E)$, $|V| = n$, $|E| = m$, represent the vertices $u_i \in V$ and the columns represent the edges $e_j = (v_j, w_j) \in E$; $m_{ij} = 1$ if $u_i \in \{v_j, w_j\}$ and zero otherwise.

It is not required that G is connected, so calculating the arithmetic average value of $d_s(u,v)$ is pointless — for disconnected graphs, some distances may be infinite. Thus Marchiori and Latora define the connectivity length as the *harmonic mean* of the separation distances,

$$\mathcal{D}(G) = \frac{n(n-1)}{\displaystyle\sum_{u,v \in G} d_s(u,v)^{-1}}. \tag{3.21}$$

In [90] they report that the behavior of $\mathcal{D}$ resembles that of $\mathcal{L}$ when evaluated globally, and that of $1/\mathcal{C}$ on a local scale. In a later article, Latora and Marchiori [86] replace $\mathcal{D}$ by another measure of weighted graphs, the *efficiency* $\mathcal{E}$ of a graph. They define a graph $G = (V,E)$ by two matrices: the adjacency matrix $\mathbf{A}$ and a distance matrix $\mathbf{L}$, in which ℓ_{ij} is the "physical distance" or interaction strength between vertices i and $j \in V$. Note that this is entirely different from the number of edges $d(i,j)$ on the shortest path from i to j. They combine $\mathbf{A}$ and $\mathbf{L}$ to calculate a cost matrix $\mathbf{D}$ to represent the cost of reaching one vertex from another, for example $d_{ij} = a_{ij} \cdot \ell_{ij}$. Latora and Marchiori seem to allow also other relations from $\mathbf{A}$ and $\mathbf{L}$ to $\mathbf{D}$, as long as $d_{ij} \geq \ell_{ij}$.

The *efficiency* of a pair of distinct vertices $i,j \in V$ is defined as $\epsilon_{ij} = 1/d_{ij}$. For instance, the graph could represent a communication network, in which shorter distances are traveled faster. The *average efficiency* of the graph G is naturally defined as the average of the individual efficiencies over all $n(n-1)$ ordered pairs of distinct vertices:

$$\mathcal{E}(G) = \frac{\sum_{i \neq j \in V} \epsilon_{ij}}{n(n-1)} = \frac{1}{n(n-1)} \sum_{i \neq j \in V} \frac{1}{d_{ij}}. \tag{3.22}$$

This value is denoted by $\mathcal{E}_{\text{glob}}$. It can be normalized with respect to the complete graph K_n, in which $d_{ij} = \ell_{ij}\ \forall i,j \in V$, and $\mathcal{E}$ is the largest possible:

$$\mathcal{E}(K_n) = \frac{1}{n(n-1)} \sum_{i \neq j \in V} \frac{1}{\ell_{ij}}. \tag{3.23}$$

The normalized efficiency $\mathcal{E}_n(G)$ is therefore the quotient of the above values, in the range $[0,1]$: $\mathcal{E}_n(G) = \mathcal{E}(G)/\mathcal{E}(K_n)$. The corresponding local quantity is defined for the neighborhood subgraphs induced by $\Gamma(v)$, normalizing by local efficiency of $K_{\deg(v)}$. The definition is straightforward and omitted. The local efficiency of the graph is henceforth defined as

$$\mathcal{E}_{\text{loc}} = \frac{1}{n} \sum_{v \in V} \mathcal{E}(G_v). \tag{3.24}$$

Latora and Marchiori [86] define a system's *fault tolerance* by the local efficiency measure. A small-world network, in the terminology of Latora and Marchiori, has a high local and global efficiency. This characterization applies for unweighted graphs (where $\mathbf{L}$ is the unit matrix $\mathbf{J}$) as well as disconnected graphs, unlike the characteristic path length $\mathcal{L}$, which is infinite for disconnected graphs.

Efficiency $\mathcal{E}$ is a rough approximation of both $\mathcal{C}$ and $\mathcal{L}$ (properly normalized). If the graph G is considered a parallel communication network (in which all vertices "transmit" simultaneously), $\mathcal{E}_{\text{glob}}$ measures the transmission efficiency of the network, whereas $1/\mathcal{L}$ is essentially a similar efficiency measure for a sequential communication network where one packet is transmitted and delivered before another one is introduced into the system. Latora and Marchiori explain that for systems in which the differences in vertex-to-vertex distances, the elements of $\mathbf{L}$, have small variation, these $\mathcal{E}_{\text{glob}}$ and $1/\mathcal{L}$ are essentially the same. If the graph G has mostly dense subgraphs, then the local efficiency $\mathcal{E}_{\text{loc}}$ is close to $\mathcal{C}$.

Latora and Marchiori [86] have plotted $\mathcal{E}_{\text{glob}}$ and $\mathcal{E}_{\text{loc}}$ as functions of the rewiring probability of the WS model, finding that the resulting curves of $1/\mathcal{E}_{\text{glob}}$ and $\mathcal{E}_{\text{loc}}$ match well with those of Figure 3.4 of the WS model. The empirical work does not employ the possibility of assigning weights to the edges, but rather concentrates on studying models similar to those in [134], such as the *C. elegans* -network (introduced in Section 2.1), the Web graph (Section 2.4), and a network model of the Boston subway transportation system.

Newman and Watts [106] modify the SWS model of Section 3.2.1 to allow reflexive and multiple edges during the shortcut addition (for analytical simplicity) and exchange the underlying lattice from $C_{n,k}$ to $\mathbb{Z}^2$ and hypercubes. We call this the MSWS model. The interesting regime of graphs for Newman and Watts is that where p is small, as they believe it to be the natural regime for modeling social networks. They define the *length scale* ϵ of a small-world graph to be the typical value of the pairwise distance $d(u,v)$ for shortcut edges when the edge (u,v) itself is ignored. They shake off a factor of two for analytical simplicity, defining $\epsilon = (pkd)^{-1/d}$, where p is the shortcut probability, k is the range to which each vertex of the underlying lattice is locally connected (similar to parameter p of the Kleinberg model), and d is the dimension of the underlying lattice. Later ϵ is called the *characteristic length* of a graph by Newman, Moore and Watts [104].

If $p \to 0$, ϵ diverges and $\epsilon \sim p^{-1/d}$. Newman and Watts [106] argue that ϵ is in fact the *cross-over length* for the transition from "a large world to a small world", discussed in e.g. [17]. This cross-over in concrete terms means that the characteristic path length $\mathcal{L}$ changes from being linear in terms of the graph order to logarithmic; ϵ defines the point when this happens. Newman and Watts state that also the average number $V(r)$ of neighbors that a vertex has within a radius r from itself, can be expressed in terms of ϵ as $V(r) = \frac{\epsilon(e^{4r/\epsilon}-1)}{2}$. Together with Cristopher Moore, Newman and Watts [104] also give a mean-field approximation to the path length distribution of MSWS graphs. The approximation is exact as the lattice gets large, $n \gg 1/kp$.

Most of the above models achieve small-world characteristics by combining randomness and regularity, but Kasturirangan [74] argues that the "fundamental mechanism behind the small-world phenomenon" is the presence of edges of several different *length scales* and therefore graph constructions that introduce multiple length scales can achieve the small-world property. He defines the length-scale of a newly introduced edge as the distance between the vertices it connects if the edge in question was not present in the network. Thus the distribution of length scales in a set of new edges to be added to an existing network is defined by the current distances of their endpoints. Kasturirangan defines that a graph $G' = (V, E \cup E')$ obtained from $G = (V, E)$ by adding the edges in E' is *multiple scale with respect to* G if the length-scale distribution of E' contains $r \gg 0$ length scales ℓ_i such that $0 < \ell_1 \ll \ell_2 \ll \ldots \ll \ell_r \leq |V|$. In general, a network $G = (V, E)$ is said to be a *multiple scale network* if there exists a subgraph $H = (V, E')$, $E' \subset E$, such that G is multiple scale with respect to H.

Hence according to Kasturirangan [74], the introduction of long-range edges is not as relevant in obtaining a small-world network than using a proper distribution (with sufficiently many different length scales) for the length scales of the edges introduced to the network. Kasturirangan also points out that the structure of the brain should portray small-world characteristics, which were already observed by Watts for the *C. elegans* neural network.

3.3 Scale-free networks

Not all observations of natural phenomena fit the small-world approach. For example, the degree distribution of the WS model and its variants seem to differ significantly from many natural networks. Natural networks often have some vertices of very high degree, which are absent by construction from both the Watts-Strogatz model and the Kleinberg lattice model. Also the ER model fails to match the degree distribution of many natural networks.

To illustrate this mismatch, in Figure 3.7 we have plotted the degree distributions generated by the ER, SWS and KL models both on linear and logarithmic scale. All graphs are of order 10,000. Thirty independent instances were generated for each set of parameters used. We used $p = 0.25$ for ER, $p \in \{0.25, 0.5, 0.75\}$ and $k = 100$ for SWS (from left to right in the plot), and the parameters $r = 2$, $p = 10$, $q \in \{10, 30, 50\}$ for the KL model (from left to right). The distributions shown are averages over the respective sets. For comparison we show in the rightmost column plots of a collaboration graph as an example of a natural network; it is a subgraph of a larger collaboration graph discussed in more detail in Section 5.3. Note how the distribution does not resemble any of those generated by the models.

The observation behind the *scale-free* network models is that the degree distribu-

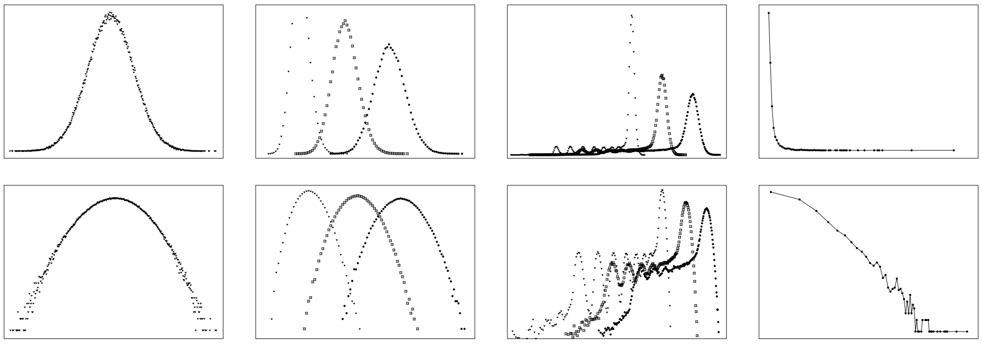


FIGURE 3.7: Shapes of the degree distributions for the ER, WS, and KL models from left to right. The rightmost figures show a collaboration graph with $n = 11{,}004$ and $m = 17{,}055$. The upper plots are drawn on a linear scale and the lower on a log-log scale.

tion of several natural networks, that is, the probability that a vertex has degree k, obeys a *power-law* $P(k) \sim k^{-\gamma}$; these distributions were introduced in Section 2.3.2. Power laws $f(x) \sim cx^{-\gamma}$ are called *scale-free* due to the fact that when x is multiplied by a constant, the proportionality of $f(x)$ to $x^{-\gamma}$ remains valid [88].

Values of the exponent γ for natural network models have been recorded eagerly since the 1999 paper of Faloutsos et al. [46]. For many natural network models the value has been observed to settle between two and three [40]. Some these exponents are listed in Table 3.2 for reference; note that most of these networks are growing natural networks and that the values of the exponents depend on the time and accuracy of the measurement. However, not all natural networks are scale-free; Amaral et al. [8] find for example that the cumulative distribution of the number of acquaintances for a network of 43 Utah Mormons resembles a Gaussian curve rather than a power-law. Although their network is very small, there is reason to believe that some particular mechanisms are required in the generation process of a network for it to obtain scale-free distributions for properties such as the degree distribution.

3.3.1 The Barabási-Albert model: Growth and preferential attachment

In 1999, Barabási and Albert [12] suggested that independent of the complex, natural network under study, the degree distribution obeys the power law $P(k) \sim k^{-\gamma}$, where $P(k)$ is the probability that a vertex has degree k. They draw the conclusion that such decay "indicates that large networks self-organize into a scale-free state" [12], which does not fit the ER or WS models where the probability of having high-degree vertices decreases exponentially with k. Barabási and Albert have

TABLE 3.2: The values for the exponents of the scale-free degree distributions of some network models. For directed models, the values for the in-degree and out-degree distributions are given individually. If the reference provides only $n = |V|$ and $\overline{k}$, we take $m = |E| = n\overline{k}/2$ and round.

Network	$\|V\|$	$\|E\|$	γ	γ_{in}	γ_{out}	Ref.
Citations	783,339	6,716,198	3			[118]
IMDb	212,250	3,054,278	2.3 ± 0.1			[13]
Internet AS	8,613	18,346	1.115			[24]
Power grid	4,941	6,596	4			[12]
Synonyms	182,853	317,658	3.25			[117]
`www.nd.edu`	325,729	889,240	2.1 ± 0.1			[12]
`www.nd.edu`	325,729	1,497,135		2.1	2.45	[117]
WWW	200 million	1.5 billion		2.09	2.72	[23]

studied those natural examples for which the data is readily available: the Hollywood collaboration graph, the Web graph, the power grid of the western United States, all of which were discussed in Chapter 2, and also a citation network of papers published in refereed journals. The scaling exponents are listed in Table 3.2; note that the directed and undirected models for the WWW that are both based on the `www.nd.edu` domain have different numbers of edges as the models have different level of accuracy: in the directed model, two edges may exist between a pair of vertices whereas the undirected version only may have one.

Barabási and Albert [12] propose another model that imitates two properties of natural networks that they consider central for the observed scale invariance: *growth* and *preferential attachment*. Growth is included as natural networks are hardly of static structure or size, whereas both of the models ER and WS assume a fixed number of vertices. Obviously, all the example networks chosen by Barabási and Albert (some of which are listed in Table 3.2) grow continuously. New papers are published and new movies are filmed, for example. In their growth, often a principle of *preferential attachment* is also inherent: as new vertices appear, they are most likely linked to those existing vertices that already have high degree. This is also intuitively understandable for all of the examples: new actors appear in the films of esteemed stars, students publish first with their supervisors and certainly cite their papers, and so forth.

It also appeals to intuition that people would put such links on their webpages that lead to already popular websites. However it is not immediately clear why the out-degree of the WWW is also scale-free. It could very well be that a page that already has a lot of links evolves to have even more links because it is essentially just a link list, whereas a page that is more of a content-providing page than a link-providing page is not very likely to gain more and more outgoing links. The origin of such power laws in the Web graph are discussed by Tadić in [126].

The generation process for scale-free networks by Barabási and Albert (the BA model) is the following. The initial graph[8] $G_0 = (V_0, E_0)$ at time $t = 0$ consists of a small initial set of vertices, $|V_0| = n_0$. At time step t, a new vertex v_t is added to V and assigned d edges; the probability that v_t is connected to $w \in V_{t-1}$ is

$$\Pr\left[(v, w) \in E_t\right] = \frac{\deg(w)}{\sum\limits_{u \in V} \deg(u)}. \tag{3.25}$$

Clearly $|V_t| = n_0 + t$ and $|E_t| = m_0 + mt$. Barabási and Albert [12] state that asymptotically G_t has a degree distribution that follows a power law with $\gamma = 2.9 \pm 0.1$ and is independent of time t. Note that if $d = 1$, the structure will necessarily grow acyclic. Barabási et al. [13] analyze the BA model with a "mean-field approach", predicting $\gamma = 3$. The graphs of the BA model resemble at least some natural networks, as the classification method of natural graphs by Vukadinović et al. [128] (briefly introduced at the end of Section 2.3) classifies a biochemical network based on gene expression data close to the BA model.

Barabási and Albert [12] study models with one of the key features missing — either growth or preferential attachment — to ensure that both are needed. They conclude that for growing graphs with uniform attachment, the degree distribution is exponential, $P(k) = be^{-\beta k}$, instead of scale free. Note that the attachment is not uniform over all vertices, but those that are present in the graph at time t. Therefore "old" vertices are likely to gain more vertices than "young". For a graph of fixed order, yet exhibiting preferential attachment upon the introduction of edges, Barabási et al. find that although the network is initially scale free, $P(k)$ will take a Gaussian form as time t increases. In [13], Barabási et al. propose some modifications to the original BA model, such as the addition of edges between existing vertices of the network and also rewiring of existing edges. They expect the model to maintain the scale-free nature as long as the modifications allow for the growth process to be the dominating factor in the dynamics of the network.

Mukherjee and Manna [99] propose a model that produces scale-free networks that have a fixed set of vertices and edges are introduced according to a specific probability distribution. Their construction starts from a $C_{n,1}$ with $V = \{v_1, v_2, \ldots, v_n\}$. During each time step $t \in [1, n]$, an additional edge is placed from vertex v_t such that $u \neq v_t$ is the target vertex of the edge with probability $\Pr[(v_t, u) \text{ added}] \sim \deg(u)^{\alpha}(t)$. For $\alpha \geq \alpha_c(n)$, the degree distribution is scale-free, although $\alpha = 1$ yields the Poisson distribution.

A nonlinear model of preferential attachment where the connection probability of vertex v to the newly added vertex is proportional to $\deg(v)^{\alpha}$, $\alpha > 0$, seems to match the observed distributions of some application data better than the models

[8] The authors do not define what the initial graph is; $E_0 = \emptyset$ is implicitly suggested. This however causes problems as the sum of vertex degrees is initially zero.

that use Equation 3.25 [68, 100]. Nevertheless, Krapivsky, Redner and Leyvraz [83] find that the linear form of Equation 3.25 is necessary to produce a scale-free topology. Furthermore they state that the power-law exponent γ can be tuned to take any value ≥ 2. This result is thus is accordance to that of Eriksen and Hörnquist [45], who argue that the *linear* preferential attachment rule of Equation 3.25 is both a sufficient and a necessary condition for the appearance of a scale-free degree distribution in a growing network.

Holme and Kim [65] notice that even though the WS model incorporates the high clustering visible in natural networks and the BA model produces the power-law degree distribution also apparent in natural examples, neither model captures both of these properties. They therefore propose a modification to the BA model to allow the adjustment of clustering to a desired level. The only change to the BA model is the addition of an extra step, triangle formation: as a new vertex u is introduced to the graph G_t, a total of d edges (u, v_i) are also added. The vertices v_i are chosen preferentially according to Equation 3.25. For each edge (u, v_i), a triangle is formed by selecting a random vertex $w \in W = \{w \mid w \in \Gamma(v_i),\ (u, w) \notin E\}$ and adding the edge (u, w) to the graph. If $W = \emptyset$, nothing is added and the process continues with the next preferential edge (u, v_{i+1}) until all d edges have been processed.

Holme and Kim tune the clustering by assigning a probability $p_{\mathcal{C}}$ for performing the triangle formation step after introduction of a new preferential edge. This controls the number of triangle formation attempts and therefore acts as a control parameter for the clustering of the resulting graph. From experiments they conclude that such graphs have $P(k) \sim k^{-\gamma}$ with $\gamma \approx 3$, and that $\mathcal{C}$ approaches a finite nonzero value as the graph grows. We have included this clustering step in our implementation of the BA model, presented in Section 5.1.4.

Walsh [132] finds it problematic that $d \leq n_0$ is required for the BA model, as the resulting graphs become quite sparse when $n_0 \ll n$. He suggests the following connection probability from a new vertex v_t to an existing vertex v_i:

$$\min\left\{1, \frac{d \cdot \deg(v_i)}{\sum_j \deg(v_j)}\right\}, \tag{3.26}$$

Applying these probabilities results in d connections per new vertex *on average*. Note that d does not need to be below n_0 to obtain well-defined connection probabilities. This proposal of Walsh is just one of the numerous suggestions and generalizations that have been made based on the BA model. We summarize some of those here to provide a quick glance to the variety of scale-free models under research.

Bollobás et al. [20] criticize the analysis and argumentation of [12, 13], providing an analysis of the asymptotic behavior of the degree distribution of a close BA variant described below. They derive the exact degree distribution $P_t(k)$ at time t

of the graph generation process for $k \leq t^{1/15}$, yielding as a consequence the result $\gamma = 3$ predicted by Barabási et al. [13] by somewhat heuristic arguments. The BA variant of Bollobás and Riordan [21] (the BR model) allows multiple and reflexive edges, which seems to be a trend in analytical models. The formulation of the BR model follows.

Denote again the number of edges added upon the addition of a single new vertex by d. First, the case $d = 1$ is considered. The graphs are a result of a random process $\{G_1^t\}_{t \geq 0}$. The initial graph can be either $G_1^0 = (\emptyset, \emptyset)$ or $G_1^1 = (V_1^1, E_1^1)$ such that $V_1^1 = \{v_1\}$ and E_1^1 contains only the reflexive edge (v_1, v_1). The graph G_1^{t-1} is modified to form G_1^t by introducing a new vertex v_t and an edge (v_t, v_i) to connect it to the existing graph structure; $v_i \in V_t$ is randomly drawn with probability $\deg(v_i)/(2t - 1)$, such that the edge (v_t, v_i) is already counted into $\deg(v_t)$ in assigning these probabilities for reasons of analytical convenience.

For $G_1^t = (V_1^t, E_1^t)$, obviously $V_1^t = \{v_i \mid 1 \leq i < t\}$, $n_t = |V_1^t| = t$, and $m_t = |E_1^t| = m_{t-1} + 1$. Graphs with $d > 1$ are generated by a generalization of the above process: $\{G_d^t\}_{t \geq 0}$ is defined as executing $\{G_1^t\}$ on a sequence of vertices v'_i, where v'_i is the vertex added to G_1^t at time $t = i$. An instance of $\{G_d^t\}_{t \geq 0}$ is constructed by identifying the vertices $v'_1, v'_2, \ldots, v'_d$ to form a single vertex v_1 (that is, all occurrences of these vertices in the set of edges are replaced with v_1), the vertices $v'_{d+1}, v'_{d+2}, \ldots, v'_{d+d}$ to form v_2, etc.

Bollobás and Riordan [21] denote the probability space of instances of the above process $\{G_d^t\}_{t \geq 0}$ for $t \in \{1, 2, \ldots, n\}$ by $\mathcal{G}_d^n$. They concentrate on studying the case $d = 1$ as the general process is defined by iterating the simple process and hence the properties of the instances of $\mathcal{G}_d^n$ can be deduced from those of $\mathcal{G}_1^n$. Most importantly, they prove the following theorem:

Theorem 3.2. [21] Let $d \geq 2$, $m \in \mathbb{Z}$ and $\varepsilon \geq 0$, $\varepsilon \in \mathbb{R}$ be constants. Then *almost every* $G_d^n \in \mathcal{G}_d^n$ is connected and $\operatorname{diam}(G_d^n)$ satisfies

$$\frac{(1 - \varepsilon) \ln n}{\ln \ln n} \leq \operatorname{diam}(G_d^n) \leq \frac{(1 + \varepsilon) \ln n}{\ln \ln n}.$$

The lower bound is obtained from the above process definition by four lemmas, but the proof of the upper bound is complicated and resorts to random pairings of integers. It takes up numerous pages in [21]. We do not attempt to summarize the analysis of Bollobás et al. here but direct the reader to the original paper. The proof steps are interesting and reveal some properties of the ensemble of the BR model, such as the fact that almost all vertices can be removed from the graph with constant probability $p < 1$ without affecting the diameter. This property of scale-free graphs will be addressed further in Section 4.2.

3.3.2 Variants of the BA model

Dorogovtsev et al. [42] define another directed BA variant in which vertices have *attractiveness* consisting of initial attractiveness $\mathcal{A}_0 \leq 0$ and the in-degree of v:

$$\mathcal{A}(v) = \mathcal{A}_0 + \deg_{\text{in}}(v). \tag{3.27}$$

The attractiveness of a vertex $v \in V$ determines the probability of placing an incoming edge at v:

$$\Pr\left[\langle u, v\rangle \in E_{t+1}\right] = \mathcal{A}(v) / \sum_{w \in V} \mathcal{A}(w). \tag{3.28}$$

An interesting feature is that Dorogovtsev et al. do not fix the origin of these edges to the vertex added at that time step. The new edges might even come from outside the current graph G_t. The initial graph is convenient to fix as one vertex and d incoming edges (again the sources of these edges are left open), although the behavior of G_t after several steps will be independent of G_0 [42].

The *age* $\mathfrak{a}(v)$ of a vertex v is defined as the number of time steps that has passed since v was introduced to the graph. Note that there is only one vertex per each value of $\mathfrak{a}(v)$ if the initial graph is chosen as above. The calculations in the analysis of the degree distribution in [42] are nontrivial, yielding that at a fixed time t, the average degree of a vertex v, $\mathfrak{a}(v) > 0$, follows a power law $\deg_{\text{in}}(v) \sim \mathfrak{a}(v)^{-\beta}$, where $\beta = 1/(1 + \mathcal{A}_0/d)$. The scaling exponent for the degree distribution is $\gamma = 2 + \mathcal{A}_0/m$. Let $\mathfrak{a} = \mathcal{A}_0/d$. For $\mathfrak{a} = 0$, Dorogovtsev et al. [42] report $\gamma = 2$ and $\beta = 1$. For $\mathfrak{a} = 1$ (which corresponds to the original BA model), $\gamma = 3$ and $\beta = \frac{1}{2}$. As $\mathfrak{a} \to \infty$, the initial attractiveness dominates and all vertices have equal attractivity, which causes the scaling behavior to break: $\gamma \to \infty$ and $\beta \to 0$.

Another aging model is suggested by Amaral et al. [8], who simply classify the vertices as *active* or *inactive* and allow only active vertices to gain new edges. New vertices are added to the network, which are the origin of these new edges. An active vertex will in time become inactive either after gaining the maximum number of links allowed or randomly with a probability from an exponentially decaying distribution. In addition to assigning an age or an activity status, the vertices may be assigned a fitness that influences their degree growth as done by Huberman and Adamic [66], but we omit discussion of this generalization.

Volchenkov and Blanchard [127] propose a stochastic model that does not employ a preferential attachment mechanism and may produce, among other types of graphs, also scale-free networks. The process starts with a graph $G = (V, \emptyset)$ at a vertex v_i. New directed edges are added from v_i to random vertices that are not yet adjacent to v_i. When the system exceeds a fluctuating stability threshold due to the addition of edges, the process moves to another vertex v_j. Therefore the out-degree of a vertex v_i is proportional to the number of time steps the process has stayed at v_i.

The in-degrees are a result of a uniform random process. The definition of stability and setting the threshold as well as controlling the fluctuations act as parameters to the model. The out-degree distribution is reported to follow a power-law and the in-degree distribution a Gaussian for properly selected parameters, but also very different network topologies may result for different parameters. The simplicity of the basic idea is appealing, but setting the parameters is nontrivial.

3.4 Combining small-world and scale-free properties

The BA model was originally suggested to explain the degree distributions observed in natural networks (as shown in Table 3.2) that differ from those produced by the earlier models, such as the ER and WS models. However, also the WS model was designed to explain properties of natural networks: small average distance combined with large clustering coefficient, which together constitute the small-world phenomenon (as shown in Table 3.1). It is unfortunate than by construction the clustering coefficient $\mathcal{C}$ of the BA model is small: as the hubs attract most of the edges, it is unlikely that dense neighborhoods are formed. Even with the clustering step of Holme and Kim, the clustering stays small for small d and hence for small δ. In this section we study recent models in which the scale-free degree distribution has successfully been combined with high clustering, which would better match the properties of natural networks. Bu and Towsley [24] propose the following generalization of the preferential probability formula of Equation 3.25:

$$\Pr\left[v_i \text{ is chosen}\right] = \frac{\deg(v_i) - \beta}{\sum\limits_{v_j \in V} (\deg(v_j) - \beta)}, \tag{3.29}$$

where $\beta \in (-\infty, 1)$. For small values of β, the high-degree nodes have less advantage of being chosen. If β were allowed to take a value $b \geq 1$, vertices with $\deg \leq b$ would not gain any new edges. Bu and Towsley show that this probability distribution also produces a power law for the degree distribution. In other respects their model is very close to the BA model, but the addition of links is altered with the addition of vertices. The initial graph contains a tree of n_0 vertices[9] and the generation method of this initial graph is not fixed. With probability p (a parameter), $d' < d$ edges are added to the graph, using Equation 3.29 to determine the endpoints. With probability $1 - p$, a new vertex with d edges is added using Equation 3.29 to determine the second endpoint. Duplicate and reflexive edges are not forbidden by construction. The analysis of the model yields

$$\Pr\left[\deg(v_i) \geq k\right] \propto k^{\frac{2d' - \beta(1-p)}{(1+p)d'}}. \tag{3.30}$$

[9] Bu and Towsley [24] require n_0 vertices and $m_0 = n_0 - 1$ edges in the initial graph, but do not explicitly state whether the initial graph is connected. For a graph with n_0 vertices and $n_0 - 1$ edges to be connected, it must be a tree.

Bu and Towsley [24] also report experiments that show their model to come closer to the measured values of the power-law exponent γ and the clustering coefficient $\mathcal{C}$ for the Internet AS level graph than other power-law generation models (including the BA model introduced in the previous section and the Inet generator of Section 2.3.3). For the characteristic path length $\mathcal{L}$, their generator is not the closest one, but the accuracy may be adjusted by allowing $\mathcal{C}$ to differ more from the measured value. Fifty graphs were generated with each generator using different random seeds. For the values measured for the Internet AS level, see Tables 3.1 and 3.2; the former table shows the most recent of the measurements reported in [24], dating at January 2002, whereas the latter contains values from September 2000. However the values of $\mathcal{L}$ and $\mathcal{C}$ have remained almost constant during the six measurements reported by Bu and Towsley: $\mathcal{L} \in [3.6168, 3.6367]$ and $\mathcal{C} \in [3.5585, 3.7914]$ in all of the measurements.

Davidsen, Ebel and Bornholdt [33] formulate a model that they have found to generate graphs with high clustering in terms of $\mathcal{C}$, small average distance $\mathcal{L}$ and scale-free degree distribution. Their model is defined in terms of an acquaintance network, where a common acquaintance may introduce two strangers. The graph $G = (V, E)$ evolves by the addition of new vertices or the replacement of existing vertices. With probability p at each time step, a random vertex v is removed from the network along with all of its incident edges, and replaced by a new vertex that only has one random acquaintance. At each time step a vertex $v \in V$ and two of its neighbors $u, w \in \Gamma(v)$ are picked randomly. The edge (u, w) is added to E unless it is already included. If v has only one neighbor, v is connected by an edge to a random vertex $w \in V$. We presume that reflexive and multiple edges are implicitly forbidden, as they are not semantically justified for acquaintance networks. The order $n = |V|$ of the graph stays fixed as such steps are iterated.

Davidsen et al. [33] concentrate on the regime $p \ll 1$, where the linking process dominates the death process and a power-law degree distribution appears. If $p \approx 1$, the random linking of the replacement vertex will dominate and the Poissonian distribution of these random links influences the degree distribution. They state that in stationary state of the generation process, $\mathcal{C} = 1 - p(\overline{k} - 1)$, where $\overline{k}$ is the average degree. Hence the clustering is considerably larger than the corresponding value of a $G_{n,\frac{n\overline{k}}{2}}$. They also derive an estimate to $\mathcal{L}$, which is of the same magnitude than the corresponding value of a random graph, and calculate the power-low exponent γ to be 1.35 for $p = 0.0025$ (the presence of the death-process causes an exponential cutoff in the distribution).

3.5 Deterministic network models

The graph models discussed thus far have included a random element. The ER model is a simple stochastic process that adds edges between n vertices, whereas in the small-world models the existence of some edges may be predetermined and the connection distribution modified (as in the KL model). The BA model of scale-free networks relies on stochastic growth, introduction of shortcuts and preferential attachment. The elimination of such stochasticity would be of great theoretical interest, as true randomness is nearly impossible to obtain and stochastic systems are often harder to understand than deterministic algorithms [15, 40].

Comellas, Ozón, and Peters [29] present a deterministic model for creating graphs of high clustering and small diameter that conforms to exact graph-theoretical analysis. They justify using $\mathrm{diam}(G)$ instead of the average or characteristic path length $\mathcal{L}$ by ease of calculation. Another change with respect to the previous models is the regularity of the resulting graphs; they add edges similar to the shortcuts of WS model to reduce the diameter, but restore the degree of vertices by edge replacement. The model is different from those previously discussed as it is *deterministic* instead of probabilistic. Deterministic small-world networks are also discussed in Ozón's doctoral thesis [110].

The starting point of the modification process is again a circulant graph $C_{n,k}$. The diameter of the original graph is therefore given by Equation 3.9, although in [29] $\mathrm{diam}(C_{n,k}) = \lceil n/2k \rceil$ is being used, which is only valid for even n; for odd n the diameter is exactly $(n-1)/2$ as the longest distance from a vertex v to any of the other $n-1$ vertices is the distance to the farthest vertex on both "sides", and there are exactly $(n-1)/2$ vertices per side. See Figure 3.3 for an illustration where $\mathrm{diam}(C_{5,1}) = 2 \neq \lceil \frac{5}{2} \rceil = 3$.

Comellas et al. [29] reduce the diameter of the circulant graph to a desired value $\mathrm{diam}(H)$ by using a graph H of that particular diameter[10] to connect $|H| = h$ vertices in $C_{n,k}$. They call these h vertices *hubs*. This will temporarily cause the modified graph G to lose k-regularity, but this will be corrected during the second step of the modification process. The only parameter of the model is the number of hubs h. In addition to shrinking the diameter, Comellas et al. analyze the clustering properties of the resulting graph G. They first calculate the clustering coefficient of the circulant graph $C_{n,k}$ (Theorem 3.1). As each vertex $v \in V$ participates in $\mathcal{T}_v$ triangles, and $\mathcal{T}_{C_{n,k}} = \frac{1}{3} \sum_{v \in V} \mathcal{T}_v$. The resulting graph G needs to be modified further to restore k-regularity. The h hubs have now a degree higher than $2k$ and therefore some of the edges connected to them need to be removed. Those that connect a hub v_i to $v_{i \pm k}$ cannot be removed as this could increase the diameter.

To study the effect of edge removal from the hubs, Comellas et al. fix H to a *double*

[10] For example, $\mathrm{diam}(K_n) = 1$ and $\mathrm{diam}(K_{1,n-1}) = 2$. The latter are called *star graphs*.

loop graph $C_{n;a,b}$, $a \neq b$, which is a 4-regular graph resembling $C_{n,k}$ graphs in which each vertex is attached to the neighbors on the ringside that are either at distance a or distance b in either direction. If $a, b \neq 1$, the "ring" $C_{n,1}$ is not a subgraph of $C_{n;a,b}$. They state that

$$\mathrm{diam}(C_{n;a,b}) = \left\lceil \frac{-1+\sqrt{2n-1}}{2} \right\rceil. \tag{3.31}$$

for $a = \mathrm{diam}(C_{n;a,b})$ and $b = a+1$ (see the references of [29] for the origin of this equation). Combining $C_{n;a,b}$ with $C_{n,k}$ increases the degree of hubs by four. They state the following lemma to decrease hub degree and analyze the effect on the clustering coefficient $\mathcal{C}_G$:

Lemma 3.1. [29] *Let v_i be a vertex of $C_{n,k}$ and $R = \{v_{i-a}, v_{i-b}, v_{i+c}, v_{i+d}\}$ be an independent set. Removing first the four edges (v_i, v_{i-a}), (v_i, v_{i-b}), (v_i, v_{i+c}), and (v_i, v_{i+d}), $0 < a, b, c, d \leq k$, thereafter adding the edges (v_{i-a}, v_{i+c}) and (v_{i-b}, v_{i+d}) reduces the number of triangles $\mathcal{T}$ in $C_{n,k}$ by $4k-6$, maintaining the degree of vertices in R and reducing $\deg(v_i)$ by four.*

Proof. Select $R = \{v_{i-a}, v_{i-b}, v_{i+c}, v_{i+d}\}$ such that $1 \leq a, b, c, d < k$, $a \neq b$, $c \neq d$. According to the previous proof, the number of triangles removed on one side of v_i is $2k-(a+1)+2k-(b+1)-1$ (a triangle is counted twice, therefore the subtraction of one). For the other side respectively $2k-(c-1)+2k-(d-1)-1$. Therefore a total of $8k-(a+b+c+d)-6$ triangles have been removed due to the removal of four edges.

The addition of two new edges naturally introduces new triangles. The number of common neighbors of the newly connected nodes, $|\Gamma(v_{i-a}) \cap \Gamma(v_{i+c})| = k-a+k-c = 2k-(a+c)$, is same as the number of new triangles in the resulting graph. Also the other new edge contributes $k-(b+d)$ new triangles. Therefore the total reduction in the number of triangles in G is $(8k-(a+b+c+d)-6)-(4k-(a+b+c+d)) = 4k-6$. This does not depend on the choice of R as long as no duplicate edges are added. □

We also summarize the derivation of $\mathcal{C}(G)$ of the modified graph:

Proposition 3.3. [29] *For a graph G produced from $C_{n,k}$ by adding $h \geq 8$ hubs and removing edges as described above to restore regularity, the clustering coefficient is*

$$\mathcal{C}_G = \mathcal{C}(Cn,k) - \frac{6h(2k-3)}{nk(2k-1)} = \frac{3(k-1)}{2(2k-1)} - \frac{3h}{nk(2k-1)}.$$

Proof.

$$\mathcal{C}_G = \frac{1}{n}\sum_{v \in V} \mathcal{C}_v = \frac{1}{n}\sum_{v \in V} \frac{\mathcal{T}_v}{k(2k-1)} = \frac{3\mathcal{T}_G}{nk(2k-1)} \tag{3.32}$$

by the definitions of the clustering coefficient and the number of triangles in a graph. It is also known from the previous lemma that $\mathcal{T}_G = \mathcal{T}_{C_{n,k}} - 2h(2k-3)$, and hence

$$\mathcal{C}_G = \frac{3(\mathcal{T}_{C_{n,k}} - 2h(2k-3))}{nk(2k-1)} = \mathcal{C}(C_{n,k}) - \frac{6h(2k-3)}{nk(2k-1)}, \tag{3.33}$$

by Theorem 3.1 and the definition of the clustering coefficient. □

The construction of Comellas et al. therefore produces graphs with small (adjustable) diameter and a high clustering coefficient. They examine as an example the case where $n = 1{,}000$, $k = 5$, $h = 50$ and find that diameter of the resulting graph G is only 9% of $\mathrm{diam}(C_{n,k})$ while $\mathcal{C}_G$ is as high as 93% of $\mathcal{C}(C_{n,k})$.

Also deterministic scale-free models have been proposed that resort to hierarchical generation to obtain deterministic growth while maintaining a scale-free degree distribution. Ravasz and Barabási [117] in fact argue that the "fundamental discrepancy between models and empirical measurements is rooted in a previously disregarded, yet generic feature of many real networks: their hierarchical topology." Barabási, Ravasz and Vicsek [15] use a fractal-like approach (the BRV model) illustrated in Figure 3.8 to construct hierarchical graphs in an iterative manner. The initial graph G_0 consists of only one vertex r_0, the permanent *root* of the hierarchy. At the second step, two copies of G_0, that is, two single vertices, are added and connected to the root to form G_1. The third step adds two copies of G_1 into the graph and connects each of the leaves of the copies to the root. The process continues like this, taking two copies of G_t and connecting the leaves of the copies to the root of the original G_t to obtain G_{t+1}.

By observing the process, it is simple to obtain some properties of $G_t = (V_t, E_t)$ such as $|V_t| = 3^{t-1}$ and $|E_t| = 3|E_{t-1}| + 2^t$, which simplifies to $|E_t| = 2 \cdot 3^t - 2^{t+1}$. This is because all edges are multiplied into the three copies of G_{t-1} that become subgraphs of G_t and the 2^t leaf vertices of the copies are all connected to the root vertex. Therefore, the degree of the root grows by 2^n at each step, being zero at $n = 0$, giving $\deg(r_t) = \sum_{k=0}^{t} 2^k = 2^{t+1} - 2$. The next step will produce two copies of r_n, which will no longer be the root. As only the root vertex gains additional edges, at time t there are $2 \cdot 3^{t-i-1}$ vertices with degree $2^{i+1} - 2$; namely the copies of the root vertices of time i [15].

There is a unique root in the graph, two copies of the former root, six copies of the previous and so forth. Hence Barabási et al. [15] argue that because $|V|$ grows as powers of three and $|E|$ as powers of two, the network topology is scale-free with a γ that is a multiple of $\frac{\ln 3}{\ln 2}$. The authors point out that γ can be varied by connecting a different number of the vertices of the copies into the root of the previous step. It is noteworthy that these graphs have no triangles by construction (that is, K_3 subgraphs), which yields $\forall t\ \mathcal{C}(G_t) = 0$ (see Section 3.2.1). Hence the model fails to capture the small-world phenomenon.

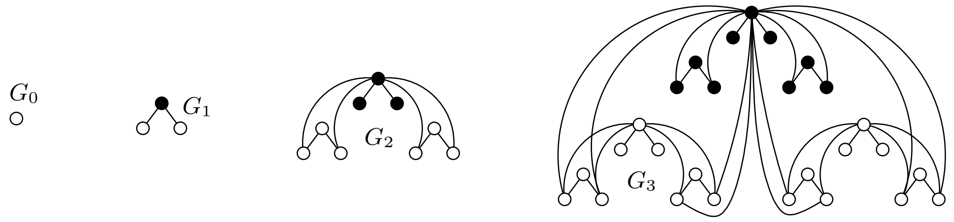


FIGURE 3.8: The "fractal" graph G_t of BRV model for $t \in \{0, 1, 2, 3\}$ (adapted from [15]). Vertices added at time t are shown white.

Ravasz and Barabási [117] propose a deterministic model to combine the scale-free degree distribution with high clustering. This RB model has the same fundamental idea as the above model. Ravasz and Barabási start with a K_5, one vertex dedicated as the *root* of the entire construction, the other vertices being peripheral (see Figure 3.9). At time $t > 0$, four copies of G_{t-1} are added to the graph of the previous step. All new vertices that are copies of peripheral vertices are marked peripheral and the peripherality mark is removed from previously peripheral vertices. After this, all new peripheral vertices are connected to the root vertex. The first three steps of the process are depicted in Figure 3.9.

As $|V_0| = 5$ and each graph G_t consists of five copies of G_{t-1}, we have $|V_t| = 5^{t+1}$. Similarly, $|E_t| = 5|E_{t-1}| + 4^{t+1}$ as four copies are always taken from each "white" vertex and all edges in G_{t-1} are copied five times to G_t. This recurrence simplifies to $|E_t| = 26 \cdot 5^t - 16 \cdot 4^t$. The degree of the root at time t is $\deg(r_t) = 4 \cdot \deg(r_{t-1}) + 4$, which simplifies to $\deg(r_t) = \frac{4}{3}(4^{t+1} - 1)$. By construction, the root vertex r_t has $\frac{1}{4}\deg(r_t)$ separate K_4 subgraphs in its neighborhood, yielding the clustering coefficient for the root vertex at time t:

$$\mathcal{C}_{\mathcal{R}}(t) = \frac{\binom{4}{2}\deg(r_t)}{4\binom{\deg(r_t)}{2}} = \frac{9}{4^{t+2} - 7}. \tag{3.34}$$

At time $t > 0$, there are $\mathcal{R}_t(i) = 4 \cdot 5^{t-i-1}$ copies of r_{t-i} in G_t for $i \leq t-1$; in total there are

$$\mathcal{R}_t = \sum_{i=0}^{t-1} 4 \cdot 5^{t-i-1} = 5^t - 1 \tag{3.35}$$

copies of former roots present at time $t > 0$. There are also $\mathcal{P}_t = 4^{t+1}$ peripheral vertices in G_t at time t, as there are originally four of them, and four copies are taken at each time step. For each peripheral vertex u_t at time $t > 0$, $\deg(u_t) = 4+t$, the induced subgraph of $\Gamma(u_t)$ has $\binom{t+1}{2}$ less edges than a clique of the same size, as the new root is not connected to any of the old root copies. This that for a vertex u_t that is peripheral at time $t > 0$,

$$\mathcal{C}_{\mathcal{P}}(t) = \frac{\binom{4+t}{2} - \binom{t+1}{2}}{\binom{4+t}{2}} = \frac{6(t+2)}{(t+3)(t+4)}. \tag{3.36}$$

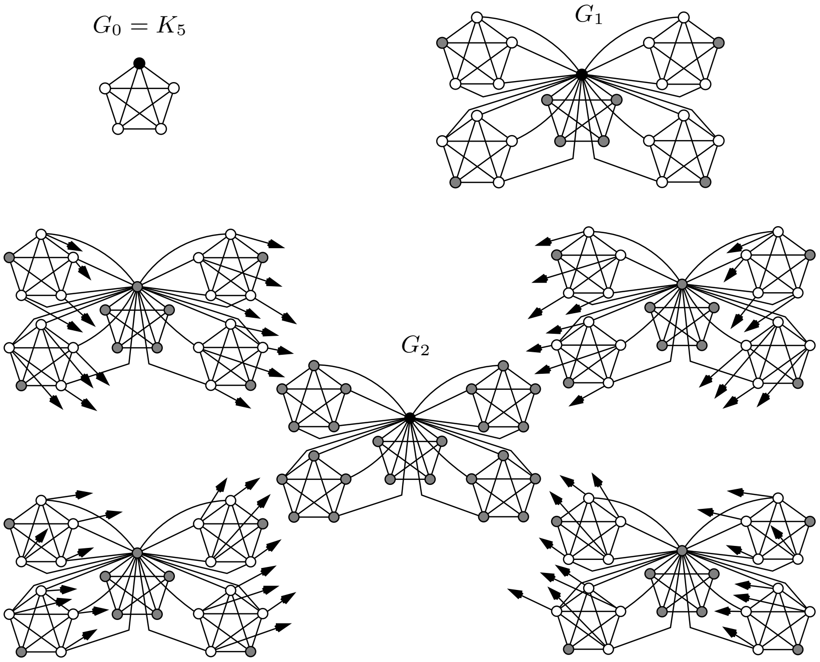


FIGURE 3.9: The graphs G_0, G_1, and G_2 of the RB model (adapted from [117]). The root vertex is drawn black, peripheral vertices white and other vertices (previously peripheral or copies of the root) are shown gray. The G_0 copies in G_1 are tilted in order to ease the root linking of the peripheral vertices. As the number of root connections in G_2 is large, only arrowheads are drawn to represent these edges.

In total there are $|V_t| - \mathcal{P}_t = 5^{t+1} - 4^{t+1}$ nonperipheral vertices, one of which is the root and $\mathcal{R}_t = 5^t - 1$ are copies of former roots. Hence there are $4(5^t - 4^t)$ such nonperipheral vertices at time t that are not the current root or copies of the former. At time t, four copies of each of the peripheral vertex of the previous step are taken, after the induced subgraph of their neighborhoods are fixed. Hence their clustering at time t is $\mathcal{C}_p(t-1)$. Also all the non-peripheral vertices of the previous step remain unaltered and four copies are taken from each. The number of *new* non-peripheral vertices at time t is by construction $\mathcal{N}_t = \mathcal{P}_{t-1} = 4^t$. Also five copies of all non-peripheral vertices of G_{t-1} are present in G_t. Note that the clustering of non-peripheral vertices is the same than their peripheral ancestors: $\mathcal{C}_{\mathcal{N}}(t) = \mathcal{C}_{\mathcal{P}}(t)$ (given in Equation 3.36).

We derive $\mathcal{C}(G_t)$ by inserting the above formulas to the definition $\mathcal{C} = \frac{1}{|V|}\sum_{v\in V}\mathcal{C}(v)$:

$$\mathcal{C}(G_t) = \frac{1}{|V_t|}\left(\mathcal{C}_{\mathcal{R}}(t) + \sum_{i=0}^{t-1}\mathcal{R}_t(i)\mathcal{C}_{\mathcal{R}}(i) + \mathcal{P}_t\mathcal{C}_{\mathcal{P}}(t) + \sum_{j=0}^{t-1}5^j\mathcal{N}_{t-j}\mathcal{C}_{\mathcal{P}}(t-j-1)\right). \quad (3.37)$$

Note that for $t = 0$ the sums are empty as the end index is lower than the start index. Our implementation of the model agrees with the above equation, but attempts to simplify this equation either by hand or with the help of symbolic software were in vain. Following the example of Ravasz and Barabási [117], we resorted to numerical simulation: as $t \to \infty$, it appears that $\mathcal{C}$ approaches 0.74184, which we have verified for $t \in [35, 440]$. For $t > 440$, the graph grows so large that the calculation of the above formula becomes infeasible ($|V_{440}| \approx 1.76 \cdot 10^{308}$). An implementation of this model is described in Section 5.1.5, where we also compare our observations on the generated topology to those of Ravasz and Barabási.

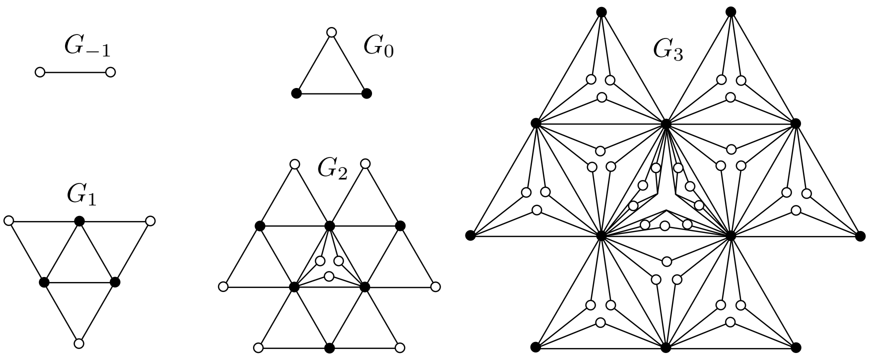


FIGURE 3.10: The pseudofractal graph G_t for $t \in \{-1, 0, 1, 2, 3\}$ (adapted from [40]). Vertices added at time step t are shown white.

Dorogovtsev, Goltsev, and Mendes [40] also question the necessity of randomness in constructing scale-free graphs. Their deterministic procedure (the DGM model) is based on [15]. The initial graph $G_{-1} = (V_{-1}, E_{-1})$ consists of two vertices v and w and the edge (v, w). At each discrete time step $t \geq 0$ of the process, per each $(u, v) \in E_{t-1}$, a new vertex w is added together with edges (u, w) and (v, w). Thus at time $t = 0$, G is a triangle. See Figure 3.10 for an illustration of the five first generations . Note that G_t remains planar[11] at each iteration. An implementation of this model is described in Section 5.1.5.

At time t, the number of edges $|E_t| = |E_{t-1}| + 2|E_{t-1}| = 3|E_{t-1}|$ is equal to 3^{t+1}, as $|E_{-1}| = 1$. Similarly, the number of vertices $|V_t| = |V_{t-1}| + |E_{t-1}|$ is equal to $3(3^t + 1)/2$. Therefore the average degree of of the resulting graph G_t is

$$\bar{k}_t = \frac{2|E_t|}{|V_t|} = 4(1 + 3^{-t}). \tag{3.38}$$

The degrees of the vertices are well-behaved: the vector $\boldsymbol{k}$ of distinct degree values at time $t \geq 0$ is clearly

$$\boldsymbol{k} = (k_1, k_2, \ldots, k_{t+1}) = (2, 2^2, 2^3, \ldots, 2^t, 2^{t+1}). \tag{3.39}$$

[11] A graph is planar if it is possible to draw a diagram in which no two edges cross. Planarity yields some interesting results and therefore can be a valuable asset for a generation model.

Denoting $\eta_i = |\{v \mid v \in V_t, \deg(v) = k_i\}|$, the vector $\boldsymbol{\eta}$ of the number of vertices with degree k_i

$$\boldsymbol{\eta} = (\eta_1, \eta_2, \ldots, \eta_{t+1}) = (3^t, 3^{t-1}, 3^{t-2}, \ldots, 3^2, 3, 3). \tag{3.40}$$

The construction follows preferential attachment, as vertices receive new neighbors proportionally to their degree. As n_t decreases as a power of k_t, the graph is a scale-free network. Dorogovtsev et al. [40] use a cumulative distribution from the above sequences $\langle k_i \rangle$ and $\langle \eta_i \rangle$:

$$P_{\text{cum}}(k) \equiv \sum_{i \geq k} \frac{\eta_i}{|V_t|} \sim k^{1-\gamma}, \tag{3.41}$$

where $\gamma = 1 + \frac{\ln 3}{\ln 2} \approx 2.585$, which falls in the range $(2,3)$ as desired. The authors of [40] also point out that the maximal degree of G_t is $\Delta = 2^{t+1} \sim |V_t|^{\ln 2/\ln 3} = |V_t|^{1/(1-\gamma)}$, coinciding with a cutoff relation for scale-free networks given in [41]. Dorogovtsev et al. also state that the number of vertices with $\mathcal{C}(v) = \frac{1}{2} k_i$ is η_i (see Equations 3.39 and 3.40 for $\boldsymbol{k}$ and $\boldsymbol{\eta}$). From this observation, it is straightforward to calculate the average clustering coefficient for G_t of the DGM model from the above local result $\mathcal{C}(v) = \frac{1}{2} \deg(v)$, taking several simplification steps:

$$\mathcal{C}(G) = \frac{1}{|V_t|} \sum_{i=1}^{t+1} \eta_i \frac{2}{k_i} = \ldots = \frac{4}{5} \cdot \frac{6^t + \frac{3}{2}}{2^t(3^t + 1)}. \tag{3.42}$$

As t approaches infinity, $\mathcal{C}(G) \to \frac{4}{5}$. The analysis of the average path length $\mathcal{L}$ is complicated and the details of the analysis are not published in [40], the initial result being that the average path length behaves similarly to $\ln V_t / \ln \bar{k}$, where $\bar{k}$ is the average degree of G_t as above. A plot of $\mathcal{C}$ and $\mathcal{L}$ is shown for $t \leq 13$ in Figure 3.11, together with the degree distributions. Computing $\mathcal{L}$ becomes infeasible quickly as the graph grows, as it requires calculating all pairwise distances.

We omit here comparison with $\mathcal{G}_{n,m}$ graphs with the same order and size, as the DGM graphs are quite sparse and hence the corresponding ER graphs tend to be disconnected. Therefor $\mathcal{L}$ is not properly defined for most graphs of the ensemble. In general, $\mathcal{L}_{\text{rand}} \sim (\ln n)/(\ln \frac{2m}{n})$ for $\mathcal{G}_{n,m}$, whereas $\mathcal{L}$ of the plot appears linear. Note that $\mathcal{C}$ quickly converges to $\frac{4}{5}$ as expected; the density δ corresponds to $\mathcal{C}_{\text{rand}}$. The degree distributions are scale-free and fall nicely on a line, except for the highest degree value which settles a little to the right of the line. The slope of the fitted lines is the same; fitting by `gnuplot` to the last two distributions, ignoring the last data point, yields $\gamma = 1.58469$.

Because the process somewhat resembles a fractal process, Dorogovtsev et al. have chosen to call it *pseudofractal*. They examine the effect of deleting random vertices or edges of G_t. They state that in order to eliminate the giant component from G_t, almost all vertices or edges must be removed. This is a known property of scale-free networks with $\gamma \leq 3$ [28]. They conclude that they “have failed to find any

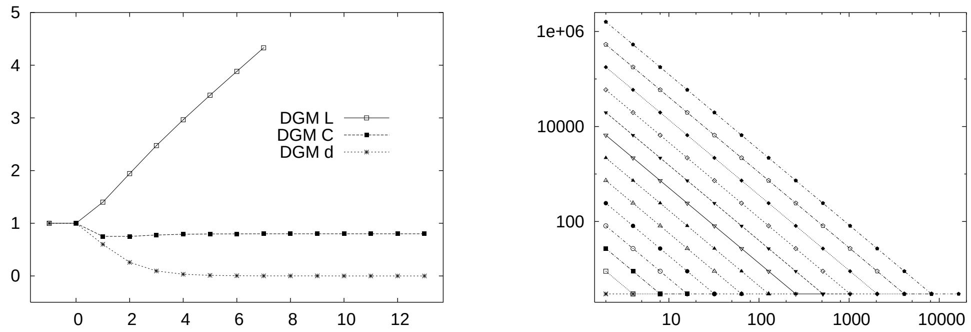


FIGURE 3.11: Values of $\mathcal{L}$, $\mathcal{C}$ and δ (left) and degree distributions (right) of the DGM graphs for $t \in [-1, 13]$. The degree distributions settle in left-to-right order such that G_{13} is the rightmost plot.

principal difference between the structural properties of pseudofractals and those of random growing nets", which somewhat surprisingly suggests that stochasticity is not absolutely necessary to adequately model natural networks [40]. In practice, the details of the application at hand will dictate whether randomness should be included.

Ravasz and Barabási [117] study common examples of natural networks, including the portion of the World Wide Web under the domain `www.nd.edu` (see Section 2.4 and [6]) and the IMDb collaboration network of actors (see Section 2.1). They find a scaling law $\mathcal{C}(v) \sim 1/\deg(v)$ to be a good approximation for several natural networks. However the above scaling law does not hold for the ER, WS and BA models or their straightforward variations, as the clustering coefficient is independent of the degree of the vertex. Nevertheless it is valid for the above DGM model, where $\mathcal{C}(v) = 2/\deg(v)$, which suggests that it is in some sense more realistic than the other models.

Yet another deterministic approach has been taken by Jung, Kim and Kahng [71]. They remark that the BRV model has fixed characteristic path length[12] $\mathcal{L}$ independent of the system size. Therefore the BRV model may be useful in application areas that also have this property (e.g., metabolic networks [69]). The model of Jung, Kim and Kahng [71] is very similar to the DGM model; it simply exchanges the triangle shape of the DGM model to a tree structure. The analysis of the model is simple, yielding solutions for the degree distribution and characteristic path length. The model also incorporates a parameter for tuning the scaling exponent within the range $2 < \gamma < 3$.

[12] Jun, Kim and Kahng [71] and many others call the average of the shortest path lengths the *diameter* of the graph, which in this text has been defined as the maximum of shortest path lengths. This unfortunate lack of consistent terminology is prevalent in applied graph theory.

Chapter 4

Properties of nonuniform random graphs

In this chapter we review some important applications of nonuniform random graphs such as epidemiology and system security and discuss algorithmic implications of the network models. The chapter ends with a study of algorithms for finding clusters in graphs, including our proposal for a local clustering heuristic.

4.1 Epidemic spreading

Studies of dynamical behavior in networks often involve problems of *spreading*, in which the propagation of some *influence* along the network structure is studied. Such influence might be for example a forest fire or heat conduction in metal. An obvious field to study spreading is *epidemiology*. If healthy individuals are infected with rate μ, and infected individuals are cured with rate δ, the *effective spreading rate* of the epidemic is the ratio $\lambda = \mu/\delta$. In many networks, such as the $G_{n,p}$ family and locally connected lattices, there exists an *epidemic threshold* λ_c below which the epidemic dies out exponentially fast and above which the epidemic spreads and remains permanently in the population [114].

Different stochastic network models of *epidemic spreading* have been defined. In the SIR model (Susceptible, Infected, Removed), an individual has a probability t_ℓ of infecting a neighboring individual, and a probability t_g of infecting a non-neighbor. Susceptibility s is the probability that a healthy individual contracts an infection when exposed to a disease, whereas transmissibility t is the probability that a healthy individual gets infected when in contact with an infected individual [97]. Infected individuals eventually die [16]. In the SIS model (Susceptible, Infected, Susceptible), individuals recover from the infection and return to being susceptible individuals instead of dying [114]. For more information on the behavior of epidemics in differently structured populations, see for example [10] and the references therein.

Watts [133] discusses epidemic spreading in the SIR model, finding that if the infection rate μ, interpreted as the probability that any infected individual will infect a susceptible neighbor, is lower than the *tipping point* $\mu_{\text{tip}} \approx \frac{1}{9}$ on WS graphs with $n = 1{,}000$ and $k = 10$, the disease only infects an $o(n)$ population before vanishing. If $\mu \gtrsim 0.5$, the disease takes over the entire population — *regardless* of the rewiring probability p. For the intermediate region, the spreading behaves differently for different values of p. Watts does not determine in detail the reasons or the nature of these differences. He also briefly examines the SIS model, where individuals are not permanently removed from the population.

As different networks allow for different spreading behavior, structural information can be useful in controlling epidemics. Pandit and Amritkar [111] use information on the far edges (discussed in Section 3.2.4) to control epidemic spreading. Assume that as a vertex is infected at time t, during the following time step $t + 1$ it will infect all of its neighbors and die. Such an epidemic will spread in a WS graph almost as quickly as in a $G_{n,p}$, due to the presence of the shortcuts [111, 134]. If the network structure is known, far edges of desired order can be computationally identified and the epidemic controlled.

Pandit and Amritkar suggest an *immunization* procedure that blocks a shortcut $e = (u, v)$ by immunizing either u or v. If i vertices may be immunized per time step, start by blocking all shortcuts and after that, immunize random vertices. If there are more than i shortcuts, the epidemic will clearly spread using the shortcuts that have not been blocked. They find that this procedure "decreases the rate of spread of the epidemic more effectively but takes longer to completely stop the spread" than completely random immunization [111].

Newman and Watts [106] use percolation (see Section 3.1.2) in small-world networks as a model of epidemic spreading, finding that the critical fraction p_c can be expressed in terms of the shortcut probability of the MSWS model (see Section 3.2.4). By solving the below equation, where p is the shortcut probability and k the radius of the local neighborhood, p_c is obtained:

$$p = \frac{(1 - p_c)^k}{2kp_c\big(1 + kp_c(1 - p_c)^k\big)}. \tag{4.1}$$

This result has been supported by numerical calculations, which however fail to match the above equation for very small values of p. This problem roots in the derivation of the equation, as is shown in [106].

Also Moore and Newman [97] study an epidemic in a small-world network $G = (V, E)$ that starts at a single individual. A graph $G' = (V', E')$ is formed on top of an underlying small-world topology as follows. A vertex $v \in V$ is present in V' if the individual represented by the vertex is susceptible to the disease whereas an edge $e \in E$ is present in E' if the disease will spread to a susceptible individual along that edge. An interesting question is, what fraction p_c of either V or E must be present in G' before a giant component appears in G? Studying this as a question on percolation both analytically and experimentally, Moore and Newman derive expressions for the threshold fractions: for an SWS graph with $t = 1$ and $s < 1$, the required fraction of vertices is

$$p_c = \frac{\sqrt{4\phi^2 + 12\phi + 1} - 2\phi - 1}{4\phi}, \tag{4.2}$$

where ϕ is the average number of randomly added edges per an edge in the underlying $C_{n,k}$ (see the end of Section 3.2.1 for details of the graph construction). For the case where $s = 1$ but $t < 1$, p_c can be solved from

$$\phi = \frac{(1-p_c)^3(1-p_c+p_c^2)}{4p_c(1+3p_c^2-3p_c^3-2p_c^4+5p_c^5-2p_c^6)}. \tag{4.3}$$

Moore and Newman [97] conclude that the presence of a single infected individual will break into an epidemic infecting more than one half of the susceptible individuals above the obtained threshold p_c, whereas only about five percent are infected below it.

Pastor-Satorras and Vespignani [114] study data from computer virus epidemics and simulations, finding that in scale-free networks no epidemic threshold exists and therefore epidemics spread quite effortlessly even when the rate of spreading is slow. Dezső and Barabási [35] study how to stop epidemics in scale-free networks by providing policies that select which individuals to vaccinate and arrive at the intuitive conclusion that making the hubs (or at least all hubs of a given degree or higher, depending on the vaccination cost) immune to the virus will more likely lift the epidemic threshold sufficiently high to delimit the spreading of the epidemic.

Computer viruses are mostly SIS epidemics, as a computer can be cleaned from the virus, but without an efficient anti-virus software, it may very well be re-infected after the cleanup. A biological epidemic may to behave more like a SIR epidemic; living organisms may develop immunity or even die. The scale-free nature of the virus-spreading networks is clear, as a virus will be more likely to infect a computer that has much traffic. Pastor-Satorras and Vespignani [114] find that only when $\gamma > 4$, instead of the typical $2 < \gamma \leq 3$, the behavior of the scale-free network under epidemic spreading will resemble that of a uniform random network such as the $G_{n,p}$ family. As the case studies in Section 2.3 show, the Internet appears to fall under the "dangerous" zone of scale-free networks, in which epidemics spread quickly and persist infinitely.

Newman et al. [102] study the spread of viruses via electronic mail; these "worms" automatically forward themselves from an infected machine to all e-mail addresses listed in the address books stored on that machine. A network of contacts listed in the address books forms a directed graph on which the virus epidemic progresses. Newman et al. study prevention and control of such epidemic outbreaks in communication networks of communities such as company intranets or university campus networks. They have obtained a sample network from a university computer system, containing approximately sixteen thousand vertices representing the users of the system. About ten thousand of these vertices are connected in the sense that a virus may be passed from one vertex onto the other. The network is not scale-free: the in-degree distribution is $P(k) \sim e^{-k/c_i}$, where $c_i \approx 8.57$ whereas the out-degree distribution is $P(k) \sim (1\sqrt{k})\exp(-\sqrt{k/d})$, where $d \approx 4.18$. According to Newman et al. [102], the former distribution form occurs in models of growing networks with random edge assignment and the latter in models with sublinear preferential attachment.

The observations of Newman et al. on address book structure imply a model in which the source of a new edge is chosen preferentially and the target randomly. This is reasonable: in the real world, some people keep address books, whereas some do not. Those that already have an address book tend to add new entries regardless of whether the person being added has a habit of keeping an address book. Implementing such a model and running tests to study the epidemic spreading in more detail would be an interesting task to consider in further work. Newman et al. find that a random immunization of vertices does not have a significant effect on epidemic spreading, whereas targeted protection of a suitably selected 10 percent of the vertices (for example by installing anti-virus software on the computers) immunizes almost the entire network. This is a promising result and complies with those obtained for attack tolerance of networks in the next section.

In conclusion, knowledge of the network structure helps to control epidemics and this observation can be harnessed to produce better practical tools for such protection. In the following section, we briefly focus on a related phenomenon of protecting networks against random breakdowns or targeted attacks.

4.2 Error and attack tolerance

It is often assumed that the high level of error tolerance demonstrated by complex natural systems is due to their apparent redundancy. Albert, Jeong and Barabási [7] argue that redundancy alone is insufficient to guarantee robustness; also a scale-free topology is required. They have experimented on natural network models, and found that a surprising amount of vertices may be disabled without affecting the interconnectivity of the remaining vertices. However, these systems are vulnerable to attacks on a few vertices of very high degree that are mainly responsible for maintaining "routes" between vertices of lower degree.

Random breakdowns can be imposed on a connected graph by removing randomly chosen edges or vertices. Vertex removal of course results in the loss of all incident edges as well and is therefore more severe. Considering vertex removal, we denote the fraction of vertices being removed by p, which means that pn vertices will be lost and therefore the probability for a single vertex to "break down" is p. After a sufficient number of such removals has taken place, the graph is no longer connected and the size of the largest component is no longer of order n. The fraction of vertices (together with the incident edges) that need to be removed for this to happen is the critical threshold p_c; the graph remains connected with high probability only if $p \leq p_c$.

Albert et al. [7] study changes in the diameter of graphs for varying values of p. They compare the ER model, which has a degree distribution with an exponential tail (that is, most vertices have similar degree), and the BA model with a scale-free degree distribution, finding that the diameter of an ER graph grows monotonically with p even though the edge set of such a graph is hardly minimal for keeping the graph connected. As the connection probability grows, the random graph becomes redundant in the sense that multiple paths connect almost any pair of vertices. For the BA graphs, the diameter remains practically unaffected as p grows from zero to 0.05.

It is nevertheless intuitive that if there are very few vertices with extremely high degree, often called hubs, they are unlikely to be among the five percent of removed vertices; having all of them removed is highly unlikely. Unlike random removal, a systematic *attack* against these hubs will have drastic consequences on a BA graph whereas for ER graphs, random breakdown and disabling the vertices in order of decreasing degree do not substantially differ. Albert et al. [7] find that $p_c \approx 0.28$ for the ER model under both attack strategies, whereas for the BA model, $p_c \approx 0.18$ for the systematic attack on hubs and p_c is close to one under random breakdowns.

Albert et al. [7] analyze the behavior of the Internet and the Web graph, which both hold up to almost 100 percent of random breakdown but fall quickly and abruptly under systematic attack. The authors estimate $p_c \approx 0.03$ for the Internet and $p_c \approx 0.067$ for the Web graph under systematic attack on hubs. This is a serious vulnerability for these and other communication networks in which the hubs are easily identified. Shargel et al. [120] propose a model that produces graphs with degree distribution varying from exponential to scale free and study the robustness of the resulting networks, finding that neither one of the extremes produces the optimal network. We refer the reader to [120] for details of the study.

Random vertex removal can be considered a percolation phenomenon as in the studies of Cohen et al. [28] on the resilience of the Internet. Considering networks where the connection probability of two vertices depends only on their degrees, they achieve an analytic result for the threshold p_c by ignoring cycles. Cohen et al. [28] conclude that for any graph whose degree distribution obeys the power law

$P(k) = ck^{-\gamma}$ with $\gamma < 3$, the threshold p_c approaches one as n grows infinite. Therefore fragmentation into small components does not take place. Hence the Internet, for which $\gamma \approx 2.5$, will have a very high threshold; if it were truly infinite, it would hold up under arbitrary removals, but as it is finite (yet huge, with $n > 10^6$), more than 99 percent of the vertices would have to be removed in order to fragment the giant component originally present [28]. Flajolet et al. [48] introduce an analytic measure of *network robustness* and study the robustness properties of the $\mathcal{G}_{n,p}$. They define *ℓ-robustness* as follows [48]:

Definition 4.4. A triple (G, v, w), where $G = (V, E)$ is a graph and $v, w \in V$, is ℓ-robust if and only if there exist at least two edge-disjoint paths of length ℓ from v to w in G.

They provide results for the expected number of such paths between two vertices of a $G_{n,p}$ and the threshold probability for their existence. The analysis employs generating functions. They find e.g. that any fixed pair $\{v, w\} \subseteq V$ is "likely" to be ℓ-robust if

$$p \geq p_{c_1} = \frac{1}{\sqrt{2}n^{1-\ell}}, \tag{4.4}$$

where "likely" means that the average number of edge-disjoint paths is at least one as $n \to \infty$. On the other hand, they show that "almost" all pairs $\{v, w\} \subseteq V$ are robust if

$$p \geq p_{c_2} = \frac{2}{n^{1-\ell}(\log(n^2 \log n))^{\ell}}, \tag{4.5}$$

where "almost" means that the probability for an arbitrary pair to be ℓ-robust tends to one as $n \to \infty$. It would be interesting to compare these bounds for the $\mathcal{G}_{n,p}$ ensemble to measures calculated for other models.

4.3 Optimization problems

When a network model is constructed, the main goal is to obtain some properties of the studied phenomenon on the basis of the model. The variety of problems that are of interest is wide, and many of the common problems are computationally challenging. Also the algorithms for extracting properties of graphs are numerous. Exact algorithms can be tedious to carry out, and hence approximation algorithms are often employed. The properties of both approaches are of practical importance considering issues of complexity, approximation ratio, and behavior of the algorithms on special classes of input.

For a compact review on the properties of the $\mathcal{G}_{n,p}$ ensemble with respect to some of the central graph problems, see Frieze and McDiarmid [50], who state several intriguing research problems related to the analysis and design of graph algorithms.

For information on graph algorithms in general, see for example "*Graphs, Networks and Algorithms*" by Dieter Jungnickel [72] and the references therein. In this chapter we aim to provide a brief review of the research currently conducted on the nonuniform network models. Some related experimental work is documented in Chapter 5.

When working with graph models, many application problems involve *optimization* and *search*. In optimization, the goal is to find the best possible substructure satisfying given criteria, operating under some *fitness function* that associates to each substructure a number that indicates the fitness of that substructure. Such optimization is often computationally demanding, as the number of substructures to examine may well be exponential in the instance size. Therefore *approximation* is widely used. The goal of an approximation algorithm for an optimization problem is to find efficiently a solution that differs no more than a fixed factor in fitness from the globally optimal solution. It is a matter of application to define how much the approximate solution may differ from the optimum to be acceptable. Optimization is rarely done by picking random solutions and examining their fitness, but rather systematically by a *search algorithm*.

The goal of a search is either to find the solution with the best possible fitness value or, in applications other than optimization, to determine the presence or absence of the desired property. A *complete search* examines every possible solution, which is not always feasible as it often requires exponential time or even exponential space. As a compromise, approximate solutions are computed with *heuristic* methods. A heuristic is essentially a rule for deciding where the search should proceed, designed to limit the search space without severe loss of accuracy. Many heuristic approaches involve randomization. When the entire search space is not examined or specifically pruned such that the optimal solution is found with certainty, the search algorithm is said to be *local*. A local search may very well not find the global optimum but rather a locally optimal solution.

For many types of computational problems, the substructure space can be formulated as a graph in which the possible solutions are the vertices and a connection appears if the solutions are somehow related in a meaningful sense. In a graph, the search proceeds from one vertex to another by traversing through the edges of the graph. As the goal of the search is to find the vertex with optimal fitness, some fitness function is imposed on the graph. *Branch-and-bound* is a search method that does not examine parts of the search space that cannot contain a feasible solution by bounding the fitness function. At a vertex v that has many feasible neighbors, such an algorithm selects one and recursively studies it either completely, or stops when solutions with fitness better than already obtained can no longer be found. Then the search returns to the branching vertex v and proceeds to search one of the other possibilities, never returning to a part of the search space it has already examined.

Popular local search methods include *hill-climbing*, where the search proceeds to a *random* neighbor with higher fitness than the current vertex (bound to get stuck on any vertex that has higher fitness than all of its neighbors) and *simulated annealing* that allows with a decreasing probability the search to proceed also to inferior neighbors. The *greedy* heuristic always proceeds to a neighbor with the highest fitness. Other approaches include for example *tabu search*; see [1] for a comprehensive study of local search methods. These methods may be employed with several fitness functions, using *restarts* to cover several areas of the search space. Usually, a local search finds a local optimum quickly and can be run several times for improved performance, while it still takes considerably less time than a complete search. However the quality of the solution found by local search may be poor.

In some cases, a complete search is impossible to conduct. The running time of a complete search can also vary from instance to another; when a new problem instance is begin searched, it may require exponentially more time than any preceding instance. Gomes, Selman and Kautz [59] address this phenomenon of *heavy-tailed cost distributions* by applying some randomness into complete search algorithms. For backtracking algorithms they suggest randomization in selecting among the equally promising branches, or if the fitness function is injective, among choices that receive a score above a threshold (e.g. ≥ 90 % of the maximum score). Completeness is ensured by keeping record of visited branches and hence avoiding re-entry.

Gomes, Selman and Kautz [59] suggest the use of an increasing cut-off value to stop the search when it appears to be stuck at a local optimum and restarting the search. Such randomization eliminates the heavy-tailed behavior to some extent and provides significant speedup — Gomes et al. report an improvement of several orders of magnitude for hard, real-world instances. They consider tournament construction, planning, and circuit synthesis as examples.

Shortly after the publication of [134] by Watts and Strogatz, Toby Walsh [131] reported results on the behavior of search algorithms on small-world networks. He argues that "such a topology can make search problems very difficult since local decisions quickly propagate globally" [131]. This argument is based on the observation that a local property such as clustering says very little about the global structure of the graph, such as the average path length, but heuristics often resort to local information to guide the search.

Walsh conjectures that in a small-world graph, the inherent "mismatch" of the local and global properties may mislead the search. As the search takes longer, the problem instance is considered to be harder. This conjecture is related to the work of Gomes and Selman [58], who find that the presence of *perturbations* in the structure of a combinatorial search problem may seriously mislead specialized search heuristics. They present results for the Latin square[1] completion problem (which is **NP**-complete) in which perturbations are introduced by requiring the Latin square to fulfill a locally consistent initial pattern. They conclude that using tailored heuristics for some class of search problems must be carefully planned, as even minor perturbations in the structure of the problem instances can result in a drastic performance degradation.

Martin Weigt [136] has studied the dynamics of heuristic optimization algorithms considering the VERTEX COVER problem as an example. The problem is to find a *vertex cover* $U \subset V$ for a graph $G = (V, E)$ such that for all edges $(u, v) \in E$, at least one of the endpoints is included in U: $\{u, v\} \cap U \neq \emptyset$. The optimization problem is to find the vertex cover with *minimum* cardinality $|U|$ over all vertex covers of G, which is an **NP**-complete problem [52]. Weigt [136] describes possible heuristics for finding the minimum vertex cover using the following idea: vertices that have many neighbors are more likely to get covered than those with only a few neighbors. The information used by Weigt's heuristics is limited to the degree of the vertex currently being considered for the cover. He finds that preferentially selecting vertices of high degree improves the performance of the heuristic optimization algorithms, providing an analysis for different types of algorithms using such heuristics.

Adamic [3] has studied the small-world properties of the World Wide Web and proposes a search engine that uses these properties among a set of search results to present them to end-users. He calls those webpages among the set of search results that have a small average distance (a small number of links to follow) to any other page in the search results *centers*; these are often link lists pointing to other pages on the same topic. The best center is the one with smallest average distance to all other pages. Adamic splits the set of search results into strongly connected components (SCC), selects the best center from the largest SCC of the Web graph, and forms a spanning tree starting from that center. A list of the centers with the largest SCC first and others in ascending order is displayed to the user performing the query; the user is expected to use the centers to browse in the corresponding SCC. Adamic [3] reports experiments for some queries and analyzes how connected the query responses are with respect to the clustering coefficient, proposing marketing uses for such information.

Another interesting problem on graphs is the MAXIMUM CLIQUE problem, in which a complete subgraph of a maximum order in a given graph is to be found. This has several interesting applications; Bomze et al. [22] discuss applications in coding theory, tiling, fault diagnosis, and pattern recognition. Their work also provides a good overview of the problem and the existing algorithms; both exact and approximation algorithms are abundant for this problem. In Section 5 we report

[1]A *Latin square* is a $n \times n$ table filled with n distinct values such that each value appears exactly once in each row and each column.

experiments conducted on the running time of a recent algorithm for Maximum Clique by Östergård [122] on selected network models.

The influence of *degree correlations* on computational complexity has also been considered by several authors, but we omit this interesting branch for brevity and direct the reader to the work of Vázquez and Weigt [130] and the references therein.

4.3.1 Shortest paths and spanning trees

Computing a *shortest path* between a pair of vertices is a central building block in many applications. For example the small-world property of small characteristic path length and high clustering cannot be detected unless some estimate on the average length of a shortest path between vertex pairs can be obtained. Also the diameter of a graph can be determined by calculating all shortest paths. A similar concept to a shortest path is a *spanning tree*, which is a subtree of $G = (V, E)$ containing all the vertices in V. If the edges are assigned costs, a minimal spanning tree is a tree with the minimum total cost.

Spanning trees of graphs are important in communication networks, where *messages* need to be delivered from one node to another efficiently. The process of determining the path to follow for each message is called *routing*. The information needed for such decisions is usually stored in *routing tables* that can be built and optimized based on path length and connection reliability. For example, if one wishes to "send a message" from vertex $v \in V$ to an arbitrary vertex $w \in V$ in a given graph G, one reasonable (although suboptimal) approach is to use a minimal spanning tree to determine the intermediate vertices to be traversed to reach vertex w. Applications for shortest path algorithms also include scheduling problems, abundant in many engineering disciplines [72].

In addition to pairwise communication, it is at times necessary to deliver a message to several nodes in a network instead of just one, which is called *multicasting*. When all of the nodes need to receive the message, the term *broadcasting* is used. When modeling communication networks, it is commonly assumed that a message may traverse one edge per time step, and any vertex v may pass a message to one of its neighbors $w \in \Gamma(v)$. Sometimes a multicast from v to $\Gamma(v)$ is assumed.

Frieze and Molloy [51] have derived an upper bound $\mathcal{O}(\ln n/n)$ to the smallest connection probability p for which a broadcast can be with high probability performed in $\lceil \log_2 n \rceil$ rounds in a $\mathcal{G}_{n,p}$ graph. In their analysis, a single round constitutes of all the vertices holding the message passing it to at most one neighbor. Finding efficient broadcast algorithms for especially scale-free networks would be of interest, as the presence of scale-free characteristics in communication networks is quite well established now.

Finding shortest paths is classically done with Dijkstra's algorithm, which is a $\mathcal{O}(n^2)$ algorithm in its basic form and $\mathcal{O}(m + n \log n)$ when implemented with a Fibonacci heap (see for example [72]). It would be interesting to construct an algorithm especially suited for small-world or scale-free networks that performs significantly better than this. Kasturirangan [74] employs his multiple-scale hypothesis, introduced at the end of Section 3.2.4, to formulate a local algorithm for the shortest path problem. The time-complexity of this algorithm is $\mathcal{O}(\log_s n)$, where n denotes the order of the graph, s is a scaling factor, and s^i are the length-scales, $i \leq \log_s n$. He assumes that all the length scales tightly cover the graph. The general idea of using the properties of network structure to improve algorithmic behavior is quite captivating, as there often is plenty of information available regarding the origin of the network on which a certain algorithm needs to be performed.

Kim et al. [75] have studied three path-finding strategies for BA graphs that are based on local information only, which is reasonable as the complete information of the global structure is often infeasible to obtain or process. The *greedy* strategy tries the neighbor with the highest degree first, the *random* strategy selects a random neighbor, and the *preferential* strategy selects a neighbor with probability proportional to its degree. The estimated diameter of the network with respect to the approximate shortest paths calculated with these search strategies over many graph realizations varies; for the random and preferential strategies it follows a power law, but scales logarithmically for the greedy strategy as well as for a global search of the shortest path.

Kim et al. also examine the effect of random attack or hub removal (as discussed in Section 4.2) to the proceeding of these strategies, finding similar behavior for all strategies and concluding that hence the attack vulnerability reported in [7] (summarized in Section 4.2) is a true topological property of the BA model.

Zhang et al. [139] have studied the effect of network structure on the performance of Freenet. Freenet is a *peer-to-peer network*, that is a special-purpose network formed on top of the Internet with dynamic topology. They find that a certain type of performance degradation is due to poorly clustered routing tables. They are able to improve the situation by introducing a cache replacement policy that aims to alter the network structure into a small-world network: most neighbors are chosen geographically close and some are required to be random far-away nodes.

Pandurangan, Raghavan and Upfal [112] also discuss properties of peer-to-peer networks, presenting a method of reducing their diameter by a preferential linking procedure. This essentially introduces small-world characteristics into the network. As such intentional introduction of the small-world property improves peer-to-peer networks, it is likely that the performance of other communication networks could be improved as well by intervening in the growth process of the network in question.

4.3.2 Coloring

A *vertex coloring* of a graph $G = (V, E)$ is a mapping[2] $f : V \to C$ that assigns a color $c \in C$ to each vertex such that $f(v) \neq f(w)$ whenever $(v, w) \in E$. The smallest $|C|$ for which such a mapping f exists for a given G is the *chromatic number* of G, denoted by $\chi(G)$. If $\chi(G) \leq k$, G is said to be *k-colorable*. Also the term k-partite is used, as the vertices colored with the same color form an independent set (see Section 2.2). The common decision problems concerning graph coloring are the following:

- Is it possible to color a given graph G with a given number of colors k?
- What is the smallest k for a given graph G such that there exists a proper coloring f?

It is **NP**-complete to determine whether a given graph is 3-colorable; the general decision problem is k-COLORABILITY [52]. The application areas of graph coloring algorithms include scheduling and other allocation tasks, such as register allocation or frequency assignment in GSM networks, studied by Javier Ozón in his doctoral thesis [110].

Walsh [131] has studied coloring of some of the DIMACS benchmark graphs (see Section 5.3) that are based on register allocation problems derived from real program code. He finds that the graphs `fpsol2i.1`, `zeroini.1`, and `mulsoli.1` are small-world networks: they have clustering coefficients $\mathcal{C} > 0.8$ whereas the characteristic path length $\mathcal{L}$ is actually somewhat smaller than for random graphs. Walsh has also performed experiments to study his conjecture, discussed earlier in this chapter, that a small-world topology makes a search problem hard. He finds that as the rewiring probability of the WS model increases, the number of vertices visited by a particular coloring algorithm by Mike Trick increases rapidly in the small-world region and drops again when the rewiring probability is above one half. The algorithm used is presumably the algorithm described in [93], but the URL to the implementation provided by Walsh has expired.

The graphs Walsh studied were very small, of order 100, but he runs the coloring algorithm for the same small-world graph 1,000 times, randomizing the vertex ordering, and plots the probability that the search visits more than a defined number of vertices against the actual number of vertices visited. From these plots Walsh finds that the distribution is heavy-tailed for $p = 0.065$: some runs were significantly slower than others within a single instance of graph coloring in a WS graph. However, random graphs and WS graphs with a lower rewiring probability

[2] Such maps $f : V \to C$ are called *vertex colorings*. Also edge colorings have been studied, see e.g. [37] for basic results on graph coloring.

$p = \frac{1}{256} \approx 0.004$ behave differently; the tails of these search cost distributions decay rapidly. Therefore Walsh predicts that also for other search problems, such as finding a Hamiltonian cycle, small-world networks will be more likely to be exceptionally hard than purely random graphs. Using randomization and rapid restarts to avoid the heavy-tailed distribution, Walsh finds that using a *geometric* restart ratio together with randomization reduces the search complexity efficiently even for small-world instances. Later in [132] Walsh experiments with the BA model, finding that randomization and rapid restarts are less effective on power-law graphs than on the WS graphs. He also states that the BA graphs are easier to color than random graphs, but display a wider spread of search cost.

Gent et al. [53] have studied coloring small-world networks by using a reduction from COLORING to the "standard" **NP**-complete problem of SATISFIABILITY (see e.g. [52]), and using a stochastic local search implementation to solve the SATISFIABILITY instances. The small-world graphs were WS graphs with $n = 100$, $k = 4$ with different rewiring probabilities; any instance with $\chi > 5$ was filtered out, as Gent et al. found that the coloring cost of those instances were at least an order of magnitude lower than the cost of coloring the other instances. They observed, among other things, that for larger values of the rewiring probability p, the local search did not need to be very greedy to perform optimally.

The results of Gent et al. are interesting: adding just a few random edges to the circulant graph made the cost of the local search jump, but adding more randomness into the graph eventually improved the performance of the local search procedure significantly below the cost of coloring the regular structure. Also, for *complete* search methods, the regular structure was easier to color than the randomized one, but the alterations were moderate in comparison to those of the local search. It appears that for both the local and the complete algorithms, the graphs that combine regular elements with random elements are hard to solve: the local search did worst when $p < 0.01$, whereas the complete search seemed to be most disturbed by $p \approx 0.1$. Gent et al. provide informative illustrations on their experiments, which were repeated for a hundred different realizations of the WS model for each value of p examined.

4.4 Random walks

A *simple random walk* is a stochastic process in which two choices are possible at each state. For example, random walk on a line, starting from the origin, proceeds at each time step one unit to the left with probability p or to the right with probability $1 - p$. Essentially the same concept is also known as the *Wiener process* and *Brownian motion*. For a review of the properties of such a process, see for example [62]. Some objects of study with random walks are, in simple terms, the position to which the walk will end up after a considerable amount of time, and the average time required to reach a certain position. Of special interest is the *mixing time* of the random walk: the time it takes until the probability that the walk is in a certain position is nearly constant. In rough terms, if this happens in time that is polynomial in the input size, the process is said to be *rapidly mixing* (see e.g. [18] for a detailed discussion).

The concept of a random walk generalizes to more than one dimension; on a plane, a random walk may go either up, down, left, or right, and so forth. Naturally random walks may also be defined on graphs. A *uniform* random walk is traverses a graph $G = (V, E)$ by proceeding to each of the $\deg(v)$ neighbors of the current vertex $v \in V$ with equal probability. In the general setup the walk is not assumed to have any other information of the graph but the (number of) neighbors of the current vertex. For directed graphs, the walk proceeds in the direction of the edges using $\deg_{\text{out}}(v)$ instead of the total degree. For analytical properties of random walks, see for example [62].

Bosiljka Tadić [123] studies how random walks with *adaptive* move strategies proceed in directed networks resembling the Web graph. The test networks have been generated by her own model of *directed* scale-free networks that grow and rearrange, using preferential attachment both in growth and rearrangement [124]. The model follows power laws for both the in-degree and the out-degree. The adaptive random walk is allowed to use *locally available* information, in particular the out-degree of the current vertex and the in-degrees of the neighboring vertices, to decide where to proceed. The edges are assigned weights so that vertices with high in-degree are more likely to be visited.

Tadić finds that for certain parameter values of her model, indicating a high degree of “rewiring” in the graph, an adaptive random walk proceeds to some fixed level of hierarchy in the graph considerably quicker than a uniform random walk. The difference in access time is some orders of magnitude [125]. Hence she concludes that such an adaptive walk can pass messages efficiently for that particular class of Web-like graphs when the degree of rewiring is large [123, 125].

Adamic et al. [5] study the behavior of search algorithms in power-law graphs such as the BA graphs. They hope to find efficient algorithms for that particular ensemble as so many natural networks have been shown to display a power-law degree distribution. They are especially interested in *distributed search* that lacks global knowledge or control, which was also the starting point of Kleinberg’s lattice model [79]. Distributed search algorithms, not requiring a central server to have complete knowledge of the network topology, are necessary in ad-hoc networks that are important in mobile communication (see [31] for an example and further references).

Adamic et al. propose a decentralized algorithm that exploits the power-law topology to make the search more efficient. Even though searching with a uniform random walk is more likely to visit high-degree vertices, they impose scaling to emphasize high-degree vertices during the search and construct a message passing algorithm based on this principle and apply variants of this to power-law graphs. The variation is mainly on the knowledge that a vertex possesses of its neighborhood while passing a message. The scaled approach approach passes messages somewhat more faster than a uniform random walk for the power-law ensemble.

It would be interesting to study as future work the possibility of random sampling the World Wide Web or other very large networks by using random walks in such a way that the level randomness can be reliably estimated. Also the mixing time of the walks should be small so that the sampling would be efficient. It appears that not much work on this topic exists yet. Deo and Gupta [34] propose a construction that attempts to derive results by a process of regularization through the addition of reflexive edges, which may not be the most practical approach.

4.5 Clustering

Clustering in general is defined to be the task of "unsupervised classification of patterns" into clusters, which is of interest in many disciplines but combinatorially nontrivial [67]. The patterns could be for example hand-written characters that need to be sorted into clusters that represent letters such that each hand-written sample gets classified as the corresponding letter. A clustering task typically comprises of the following subtasks:

1. representing the pattern in suitable form,
2. defining a measure to determine pattern proximity, and
3. grouping the patterns into clusters.

In this section, we restrict to the case of finding clusters in graphs, which corresponds to the last two steps: defining a measure that determines whether two vertices belong to the same cluster or different clusters, possibly choosing a proper threshold value for the measure, and finally performing the grouping into clusters. Jain et al [67] point out the importance of evaluating the resulting clusters, as "all clustering algorithms will, when presented with the data, produce clusters — regardless of whether the data contain clusters or not".

In the study of social networks, it is widely believed that there is a community structure, something similar to the caveman graphs of Section 3.2.3, in any society: people form communities that are dense in comparison to the connection density between different communities. Knowledge of such communities is important for example in epidemiological research. This topic is brought forward in recent work by Newman and Girvan, but traces back to the 1970s (see [55, 103] and the references therein). In bioinformatics, clustering algorithms are important for analyzing similarities in e.g. genomic sequences; once a similarity graph is formed, for example families of proteins may be found by clustering algorithms [76].

4.5.1 Global clusters

A traditional approach to global clustering (see for example [103] and the references therein) has been the *hierarchical clustering method*, in which connection strengths, such as the pairwise distances, of the vertex set are calculated and a new network is constructed by inserting edges between vertex pairs in decreasing order of these strength values. At each strength level, a clustering structure is visible as the connected components of the construction; the entire hierarchy may be drawn into a *dendrogram*, which is a tree that shows the clustering at different levels. This method works well for example for objects on the plane and their Euclidean distances, but it is not suited for simple graphs without physical distances or edge weights.

Also algorithms that search for maximal subgraphs that have a density higher than a preset *threshold* have been proposed (see for example [76] and the references therein). Without a threshold such an algorithm would search for complete subgraphs, which include K_2 and K_3, which are neither very appealing as clusters; any edge will produce a K_2 subgraph whereas K_3 is a simple triangle. Another approach was proposed by Matsuda et al. [91] consider p-quasi complete subgraphs as clusters:

Definition 4.5. A graph $G = (V, E)$, $n = |V|$, is *p-quasi complete* for $p \in [0, 1]$, if for all $v \in V$, $\deg(v) \geq p(n-1)$.

The connection probability p is given as a parameter to their algorithm. They show that it is **NP**-complete to determine whether a given graph has a 0.5-quasi complete subgraph of order at least k. Hence they conclude that approximation algorithms are the only feasible approach for locating such subgraphs [91].

Newman and Girvan [103] use dendrograms with *edge betweenness* as the splitting criteria; the betweenness of an edge is the number of shortest paths between arbitrary vertices that contain the edge in question. If there are k shortest paths connecting a pair $\{v, w\}$, each of them will have weight $\frac{1}{k}$ in calculating the betweenness measures of the included edges. The current algorithms to compute betweenness for an edge operate in $\mathcal{O}(nm)$ time. For results on betweenness distributions, see [129] and the references therein.

Newman and Girvan assume edges with high betweenness to be links between communities instead of internal links within a community: the several shortest paths passing through these edges are the shortest paths connecting the members of one community to those of another. Hence they split the network into clusters by removing one by one edges with high betweenness values. If more than one edge has the highest betweenness value, one of them is chosen randomly and removed. The removal is followed by recalculation of the betweenness values, as the shortest paths have possibly been altered. This gives an algorithm polynomial in n and m for clustering. Of course one still must decide when to stop the partitioning, just as for the hierarchical clustering method.

In some applications it is desirable to be able to observe the clustering structure resulting at different levels of hierarchy, but in some cases just a single clustering is required. Employing a hierarchical method in that situation requires setting a *threshold* to detect when to stop the hierarchical process. To avoid the problem of setting a threshold, *partitional* clustering algorithms that produce a single clustering structure for the given graph have been proposed. The partition may either be a true *graph partition*, where distinct clusters cover the vertex set, or a loose partition where some vertices may not be included in any cluster and some vertices may belong to more than one cluster, formally referred to as a *graph cover*.

A traditional method to produce a true partitioning into clusters for a graph is to take the *minimal spanning tree* and remove the edges that are "the weakest" under some measure: if the edge cost represents distance, remove the longest, and if the cost represents the bond strength, remove the cheapest. For unweighted graphs, some other measure of the "importance" of an edge needs to be derived, such as the betweenness measure used above by Newman and Girvan [103].

Hartuv and Shamir [64] describe the following clustering algorithm for undirected, unweighted graphs. They define that G is *highly connected* if the edge-connectivity $\underline{k}(G) > \frac{n}{2}$ (see Section 2.2). Such graphs have diam ≤ 2. If the graph $G = (V, E)$ is not highly connected itself, split it in two connected subgraphs H and $\overline{H}$ by removing a cut C of minimum order from E. Repeat for both of these subgraphs. Return upon finding a highly connected graph. Whenever an isolated vertex is encountered, it is not considered a cluster but simply grouped into a set of singletons.

Denoting by N the number of clusters found and by $f(n, m)$ the running time of the MINCUT algorithm for a graph with order n and size m, they bound the complexity of this algorithm by $2N \cdot f(n, m)$. Hartuv and Shamir also suggest heuristics to improve the behavior, such as preprocessing G by removing low-degree vertices, which in turn reduces the number of MINCUT iterations necessary. They mention experiments that show good performance of the algorithm for noisy gene expression simulated data.

Kim [76] uses biconnected components to cluster genome data; the articulation points themselves provide information of the application problem in addition to the clustering itself.

A clustering method that suits the unweighted and undirected graphs is proposed by Mihail et al. [94], who define clusters in terms of the *relative density* of a set of nodes $S \subset V$ in a graph $G = (V, E)$:

$$\delta_r = \frac{|\{(u,v) \in E \mid u, v \in S\}|}{|\{(u,v) \in E \mid \{u,v\} \cap S \neq \emptyset\}|}, \tag{4.6}$$

which is a measure of the fraction of the number of edges "inside" S, which we call the *in-degree* of S and denote by $\deg_{\text{in}}(S)$, and the total number of edges incident to S. A set S is a *good cluster* if δ_r is large.[3]

To find such clusters, Mihail et al. [94] resort to spectral analysis of G. Spectral methods are commonly employed, as the properties of the graph spectrum are often closely related to other structural properties of graphs. Goh, Kahng and Kim [57] have studied the spectrum created by the BA model for $d = 2$ (see Section 3.3.1) and are able to find the exact spectrum for graphs up to 5,000 vertices and determine the first few of the largest eigenvalues for graphs of order as high as 400,000. Hence we do not expect the clustering method of Mihail et al. [94] to scale up to very large graphs such as the Web graph, although it successfully identifies clusters from the Internet AS graph (one cluster being the dominant Internet service providers in the United States).

4.5.2 Heuristics for local clusters

Some interesting graphs are either too big to fit in the main memory of an ordinary computer or impossible to obtain completely, such as the World Wide Web, from which only small snapshots are available. Therefore an approach for locating the clusters that relies on having complete adjacency information available is not always practical. We propose an algorithm that starts from a given vertex and determines a proper cluster for that vertex using local information only. Note that such an algorithm must be an approximation algorithm, as an exact algorithm by definition would take into account the entire graph.

In a given graph $G = (V, E)$, a natural definition of locally available information is the neighborhood of the vertices included in the cluster. It also makes sense for a cluster to be connected and we therefore concentrate only on the connected component containing v instead of the entire graph when finding the cluster of v.

[3] The formula of δ_r and the description on page 8 of [94] are inconsistent, but we believe the formula to be mistyped as it is counterintuitive: it compares the number of edges between S and $V \setminus S$ to the total number of edges incident to S, which should rather be *small* for a good cluster.

We denote the connected component of v as $G_v = (V_v, E_v)$, the cluster of v by $\mathcal{K}(v) = (V', E')$, and the order of the cluster by $|\mathcal{K}(v)| = \kappa$. Note that $V' \subseteq V_v$ and $E' \subseteq E_v$. The main question is how to define clusters and what function will yield clusters that comply with the definition.

The naïve definition of a cluster as a subgraph containing v with maximum density obviously fails, as any clique has density 1 and any search process would reach an optimum upon the addition of any $u \in \Gamma(v)$ to $\mathcal{K}(v)$, as the density would be $\delta(K_2) = 1$. Hence it is essential to find a fitness function that avoids getting "stuck" at small cliques containing v. The problem is avoided with the relative density of Equation 4.6, as it only reaches the value one for a clique that has no edges pointing outside the clique from any vertex.

The cluster of v should intuitively contain at least the largest clique that contains v. This is simply achieved by including all $u \in \Gamma(v)$ that are also pairwise neighbors of each other to $\mathcal{K}(v)$, but is not entirely sufficient. This is impractical as determining maximum clique order in a graph is **NP** complete. Moreover, the natural cluster of a vertex is not necessarily a complete subgraph, but rather just a "surprisingly" dense subgraph with possibly a little less than $\binom{\kappa}{2}$ edges connecting the κ vertices.

We start by defining a measure of how "surprising" the density of a given subgraph is as the probability that $\mathcal{K}(v)$ contains as many edges as it does or more. The probability that each edge is independently present is the density of the entire graph G, $\delta = m/\binom{n}{2}$, where $n = |V_v|$ and $m = |E_v|$. Therefore the probability that $\mathcal{K}(v)$ contains ℓ or more edges is defined by the binomial distribution, where $M = \binom{\kappa}{2}$, is

$$p = \Pr[X \geq \ell] = \sum_{i=\ell}^{M} \binom{M}{i} \delta^i (1-\delta)^{M-i}. \tag{4.7}$$

To base a heuristic algorithm on this observation, we take as our fitness function $\mathcal{F} = \ln(1/p)$ and maximize. The purpose of the natural logarithm is to attenuate the exponential growth of $1/p$. The initial cluster for vertex v is $\Gamma(v)$, which is modified by allowing random additions from $\{u \mid u \in V_v \setminus V',\ w \in \mathcal{K}(v),\ u \in \Gamma(w)\}$ and random removals from $\mathcal{K}(v) \setminus \{v\}$. Upon a vertex removal operation we also remove all vertices from $\mathcal{K}(v)$ that are no longer connected to v by a path. This is achieved without much effort by performing a depth-first search starting from v in G, restricting the search to vertices in $\mathcal{K}(v)$.

We ran iterated *simulated annealing* [77] on this fitness function: after each round of modifying the current cluster, we accept the new cluster candidate if it has higher fitness f' than the current fitness f, but if the fitness decreases, we accept with probability $\exp(f - f')/T$, where T is the *temperature* of the system. After each round, the temperature is decreased by a scaling factor α: $T' = \alpha T$. The initial temperature T_0, the scaling factor α, and the number of rounds are parameters of the search.

For small graphs we observed that vertices tend to select as their cluster the largest clique of the graph and the path connecting the vertex to the clique. It would be better if the "attractivity" of a large clique decreased exponentially as the distance to the clique grows. Also, the computations required for the binomial distribution can be tedious. However approximation of the binomial coefficient using e.g. the Stirling formula, Chernoff bounds, or simply the normal distribution is possible. The effect to running time and accuracy would of course need to be determined for such approximation. Another possibility is to define as a cluster a subset that has more edges connecting the vertices in the cluster than would be expected and no more edges pointing outside from the cluster than would be expected by the density of the entire graph. This approach is similar to that of relative density δ_r of Equation 4.6.

To be precise, we would want a cluster $\mathcal{K}$ to contain unusually many edges *in addition* to its spanning tree with respect to the density of the entire graph. We treat the possibly unknown and possibly very large graph G_v having density δ as a $\mathcal{G}_{n,\delta}$ graph; hence the average degree of a vertex $v \in V_v$ is $\overline{k} = \delta(n-1)$. To form a spanning tree, each vertex in the cluster $\mathcal{K}$ must use at least one of its edges to connect to other vertices in $\mathcal{K}$.

Denoting the order of the cluster by κ, the vertices in $\mathcal{K}$ have $\kappa\overline{k}$ edges, $\kappa - 1$ of which are the edges that form the spanning tree. This means that each $v \in \mathcal{K}$ is on average connected to $2 - \frac{2}{\kappa}$ other vertices in $\mathcal{K}$ by the spanning tree, and hence there are on average $\kappa - (2 - \frac{2}{\kappa})$ vertices left in $\mathcal{K}$ and $n - \kappa$ elsewhere in the graph to which v may connect to, which sums up to $N = n - 2 + \frac{2}{\kappa}$ vertices not yet connected to v. Calculating the fraction of the vertices not connected to v by the spanning tree but inside the cluster of all N possible neighbors, we obtain

$$p_{\text{in}} = \frac{\kappa - 2 + \frac{2}{\kappa}}{n - 2 + \frac{2}{\kappa}}. \tag{4.8}$$

Equivalently, the probability that $v \in \mathcal{K}$ will have an edge pointing outside of $\mathcal{K}$ is

$$p_{\text{out}} = 1 - p_{\text{in}} = \frac{n - \kappa}{n - 2 + \frac{2}{\kappa}}. \tag{4.9}$$

For a particular cluster candidate $\mathcal{K}'$ we may calculate the exact number of edges between vertices included in $\mathcal{K}'$, $\deg_{\text{in}}(\mathcal{K}')$, as well as the number of edges pointing outside from $\mathcal{K}'$, $\deg_{\text{out}}(\mathcal{K})$. Note that the $\kappa - 1$ tree edges exist for any cluster candidate and are hence excluded. For $\mathcal{K}'$ to be a good cluster, we want the fraction of nontree-edges incident to $\mathcal{K}'$ that are in-edges but not tree-edges to be larger than p_{in}. Simultaneously, we would like the fraction of edges incident to $\mathcal{K}'$ that are out-edges to be at most p_{out}, the smaller the better.

The following function obtains large values for good clusters and small for poor cluster candidates, and hence may act as a starting point for defining a fitness function for local clustering:

$$
\begin{aligned}
f(\mathcal{K}') &= \left(\frac{\deg_{\text{in}}(\mathcal{K}') - (\kappa - 1)}{\deg_{\text{in}}(\mathcal{K}') + \deg_{\text{out}}(\mathcal{K}')} \Big/ p_{\text{in}}\right) \cdot \left(\frac{\deg_{\text{out}}(\mathcal{K}')}{\deg_{\text{in}}(\mathcal{K}') + \deg_{\text{out}}(\mathcal{K}')} \Big/ p_{\text{out}}\right)^{-1} \\
&= \frac{(\deg_{\text{in}}(\mathcal{K}') - \kappa + 1) p_{\text{out}}}{p_{\text{in}} \cdot \deg_{\text{out}}(\mathcal{K}')} = \frac{(\deg_{\text{in}}(\mathcal{K}') - \kappa + 1)(n - \kappa)}{\deg_{\text{out}}(\mathcal{K}')(\kappa - 2 + \frac{2}{\kappa})}.
\end{aligned}
\tag{4.10}
$$

The fundamental idea behind this construction is to compare the fraction of both types of edges present to the probability than an edge is of that type by dividing the observed fraction by the expected fraction. Large value indicates that we observe a larger fraction than expected and a small value that we observe a smaller fraction than expected. A good cluster has a large value for the in-edges and a small for the out-edges. As we want f to obtain larger values when the in-degree fraction is surprisingly large and the out-degree fraction is small or as expected, we invert the latter and multiply. Note that n does not have to be exactly known; it can be replaced by a constant that is larger than the order of any cluster in the graph.

The function $f(\mathcal{K}')$ is not directly applicable as a fitness function as $\deg_{\text{out}}(\mathcal{K}')$ is zero when $\mathcal{K}$ is a component of the graph. This results in possible division by zero. Also, if the cluster candidate is only a tree, the numerator will be zero. For the purpose of local search the function should therefore be modified in some manner that does not change the "order of goodness" between cluster candidates, as we prefer to maximize a strictly positive real-valued continuous function. One simple modification is adding a constant to both the numerator and the denominator.

$$
f_1(\mathcal{K}') = \frac{(\deg_{\text{in}}(\mathcal{K}') - \kappa + 1)(n - \kappa) + 1}{\deg_{\text{out}}(\mathcal{K}')(\kappa - 2 + \frac{2}{\kappa}) + 1}. \tag{4.11}
$$

There are some problems with the functions of Equations 4.10 and 4.11: one has to know or guess the order n of the entire graph, and the values that the function can take are not limited to the same range for different values of n. Hence we define yet another fitness function with the same goals as the previous, but that only takes values in $[0, 1]$. We take the relative density of Equation 4.6 and multiply it by the density of the subgraph induced by $\mathcal{K}$, obtaining

$$
f_2(\mathcal{K}) = \frac{\deg_{\text{in}}(\mathcal{K})}{\binom{\kappa}{2}} \cdot \frac{\deg_{\text{in}}}{\deg_{\text{in}} + \deg_{\text{out}}} = \frac{2\deg_{\text{in}}^2}{\kappa(\kappa - 1)(\deg_{\text{in}} + \deg_{\text{out}})}. \tag{4.12}
$$

Experiments with the fitness functions of Equations 4.11 and 4.12 are described and reported in Section 4.5.2. We have attempted clustering of natural networks, regular networks, and networks generated by some of the models of Chapter 3.

For a local search that proceeds by crawling the neighborhoods of the included vertices in a large, possibly unknown *directed* graph such as the Web graph, the incoming edges of vertices in the cluster are unknown until their source vertices are first encountered. If these are included in the out-degree of the cluster, the search needs to restart upon their discovery. However, in the search for *local* clusters, we ignore vertices unreachable *from* the cluster. Hence it is reasonable to define the out-degree of a cluster to consists only of the edges pointing *from* the cluster to other parts of the graph. Note that this affects the expected out-degree of a cluster candidate and hence the fitness function of Equation 4.11 changes for directed graphs. However, the function of Equation 4.12 is directly applicable. We have recently constructed a web crawler that finds the cluster of a defined webpage using simulated annealing on the fitness function of Equation 4.12. We will present results obtained from this experiment in further work.

Chapter 5

Experimental results

We conducted a series of experiments by generating graphs with a subset of the generation models presented above and studying some of their structural and algorithmic properties. The software written for the experiments consists of a larger toolset written in C language and some additional tools written in Java; the source code for the toolset is available at `http://www.tcs.hut.fi/~satu/models/`. The graphs used in the experiments are undirected and have no weight or fitness functions imposed on their vertices or edges. Such generalizations may be considered in further work. We used one AMD Athlon XP 1600 MHz workstation with 1,024 MB of main memory running Debian GNU / Linux 2.4.20 to run the experiments.

At present our toolset can efficiently handle graphs up to some thousands of vertices and perform various different examinations. Sparse graphs are computationally more approachable than dense graphs of the same order. Some computations work well for graphs of more than a hundred thousand vertices, whereas some become infeasible already at a few thousand vertices. Random numbers are generated with Donald E. Knuth's [82] `ran_array` function.[1]

We label the vertices V of a graph $G = (V, E)$ with integers such that $V = \{0, 1, \ldots, n-1\}$. Edges are pairs of integers containing the source and target labels. A graph may be stored in three forms, which can be varied according to the task at hand: as a list of edges, as a complete set of adjacency lists, or as an adjacency matrix implemented as a bitmap. The graph analyzer consists of simple functions that take as an input a graph and calculate for output a certain measure. Many operations are implemented for both the adjacency list and the adjacency matrix representations; for sparse graphs we prefer the former, for very dense graphs, we use the the latter.

[1] Available online at `http://www-cs-faculty.stanford.edu/~knuth/programs.html#rng`.

In this section we explain the algorithms we use to calculate some of the measures used to analyze graphs. The calculations of most of the measures are straightforward to derive for a given graph, but a few are nontrivial especially for large or dense graphs. For example, the clustering coefficient is obtained by straightforward examination of the number of edges in the respective induced subgraphs and the degrees of the vertices. Note also that the definitions of some of these measures are properly defined only for connected graphs; for disconnected graphs, we concentrate on the largest connected component.

The All Pairs Shortest Paths problem (see e.g. [30]) needed for the calculation of the characteristic path length $\mathcal{L}$ is solved by the Floyd-Warshall algorithm [49] for sparse graphs ($\delta < 0.5$) using adjacency lists, and by exponentiation of the adjacency matrix for dense graphs. The toolkit also includes an implementation that uses Dijkstra's algorithm, for situations where the complete distance matrix would be too large to handle efficiently. All these approaches are unfortunately too slow for the exhaustive calculation of the average length of the shortest path, which we do by a breadth-first search that only counts the number of vertices at distance $1, 2, \ldots$ from a given vertex. These listings are analyzed by a simple Java tool to produce values of $\mathcal{L}$ and diam. This approach took about two hours to complete for a graph of order over 100,000 vertices.

A breadth-first algorithm we implemented to compute the girth g of a graph does not scale well to large degrees; hence also a depth-first version using incremental depth was implemented. This recursive procedure is not very efficient either. Especially regular graphs with many relatively small cycles are problematic. The efficiency might be improved if the vertices were classified by their role in the graph topology and the girth search would only take place for representatives of the vertex classes, but this efficiency aspect will be considered possibly in future work if the exact value of graph girth becomes relevant. Such a classification is however likely to be nontrivial.

The measurements requiring repetition are handled as in [109] and the references therein. A confidence interval with *confidence level* α contains the estimated value with probability $(1 - \alpha)$. For a random sample $(x_1, x_2, \ldots, x_N)$ from an unknown distribution, a *confidence interval* may be defined by using approximation of the standard deviation σ and Student's t-distribution. First approximate the expected value μ by the sample mean

$$\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i. \tag{5.1}$$

The estimated variance $\hat{\sigma}^2$ of $\overline{x}$ is therefore

$$\hat{\sigma}^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2. \tag{5.2}$$

The confidence interval of the expected value is $[\overline{x}-\Delta, \overline{x}+\Delta]$, where Δ is obtained from Student's t-distribution as $\Delta = t_{n-1,1-\alpha/2}\,\hat{\sigma}$. We obtain samples as long as the confidence level obtained by this method is lower than a desired value α, up to a maximum of sixty iterations. We kept the minimal number of repetitions for all experiments at 30 as the measured distributions are not necessarily normal. As in all experimental work, it is important to vary both the input instance and the size of the instance to produce reasonable results [73]. The goals and parameters of each experiment set is explained in conjunction with reporting the results.

5.1 Implemented generation models

In this section we describe the implemented generation models. The standard model for random graphs, $G_{n,p}$, is categorically included, as well as both of the "famous" models, the Watts-Strogatz model and the Barabási-Albert model. For the former, the more analytically approachable variation by Newman et al. is chosen, and for the latter, the clustered variant was implemented. An undirected version of the Kleinberg lattice model is included to provide a different regular structure for added randomization than the circulant graph of the SWS model. The deterministic RB model is implemented as well as the DGM model. Some regular structures also have their own generation procedures: complete graphs, complete bipartite graphs, flat and toroidal lattices, and circulant graphs. As the models are at times not described in literature with sufficient detail for unambiguous implementation, we explain here the assumptions and implementation details of the models.

5.1.1 Erdős-Rényi model

The naïve method to generate a random graph $G = (V, E)$ with n edges and a probability p that each edge independently appears in E would to draw a uniformly distributed random number $r \in [0, 1]$ for each pair of distinct vertices and include the corresponding edge in E if $r \geq p$. This would require $\binom{n}{2} = n(n-1)/2$ random numbers; one for each pair of distinct vertices. As pointed out by Nuutila [109], Kapidakis [73] provides a way of generating a $G_{n,p}$ in $\mathcal{O}(n+m)$ time instead of the above $\mathcal{O}(n^2)$ worst-case estimate. This is of interest of sparse graphs where $m \ll n^2$. The fundamental observation is that the n^2 random trials of the naïve method are independent *Bernoulli trials* with success probability p (a *success* being the addition of the edge). This suggests two things: the number of edges $|E| = m$ created in the trial set obeys the binomial distribution $Binom\left(\binom{n}{2}, p\right)$, and more importantly, when the number of trials before the first success is denoted by X, it applies that

$$\Pr[X = k] = (1-p)^{k-1}k, \quad k \geq 0. \tag{5.3}$$

This means that $\mathrm{E}[X] = \frac{1}{p}$. These observations combined give that X is geometrically distributed with parameter p.[2] The next edge will appear a geometrically distributed number of "steps" after the first, as the trials between them are again independent Bernoulli trials. Therefore we may construct a $G_{n,p}$ by hopping forward on the sequence of possible edges by steps of length obeying the geometric distribution *Geom*(p), skipping a pair (v, w) if $v \geq w$ to ensure that each pair is considered only once. We construct a blocking table that is initialized to forbid the hopping procedure to select a reflexive edge and to store information on what edges have been included. If the selection procedure hops to a position that is forbidden by the block list, it simply ignores that and hops again.

The $\mathcal{G}_{n,m}$ graphs are generated by the naïve method of choosing the endpoints of an edge randomly and uniformly among all vertices using two random integers, avoiding duplicate and reflexive edges, until a total of m edges have been added. The clustering coefficient and average path length of the ER model, namely this latter implementation, is studied together with the SWS model in Figure 5.1 on page 87.

5.1.2 Solvable Watts-Strogatz model

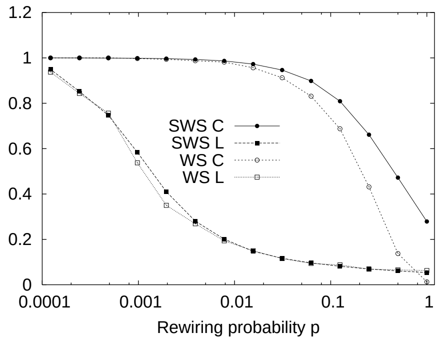
Our implementation of the SWS model, described in Section 3.2.1, is quite straightforward and uses the above $G_{n,p}$ generation procedure. First, the circulant graph $C_{n,k} = (V_c, E_c)$ is generated. Then $G_{n,p'} = (V_r, E_r)$ of the same order is generated with the probability parameter p', which depends on the parameter p of the solvable WS model. As one shortcut edge is generated per each edge *present* in the $C_{n,k}$, which by definition contains nk edges, we generate a $G_{n,p}$ where

$$p' = p \cdot \frac{nk}{\binom{n}{2}} = p \cdot \frac{2k}{n-1}. \tag{5.4}$$

As the edge generation procedure of the random graph $G_{n,p}$ takes as a parameter a blocking table that forbids certain edges (as explained above in the implementation details of the $G_{n,p}$ model), we construct such a table that forbids reflexive edges and all edges $e \in E_c$. This will ensure that the generation produces a simple graph with the property $E_c \cap E_r = \emptyset$. As the vertices of both graphs are labeled $\{0, 1, 2, \ldots, n-1\}$, the edge sets E_c and E_r may trivially be joined to obtain the SWS-graph $G = (V, E)$ such that $V = V_c = V_r = \{0, 1, \ldots, n-1\}$ and $E = E_c \cup E_r$.

We studied whether this method of generation produces graphs that fulfill the original definition of "smallworldness" by Watts and Strogatz [134] by a series of test runs. As the implementation is based on the SWS version instead of the original WS model, differences to the measurements of [134] are sure to appear:

[2] From a uniformly distributed random number $X \in (0, 1)$ we obtain a geometrically distributed random variable $Z = \lceil \ln(u)/\ln(1-p) \rceil \sim Geom(p)$.



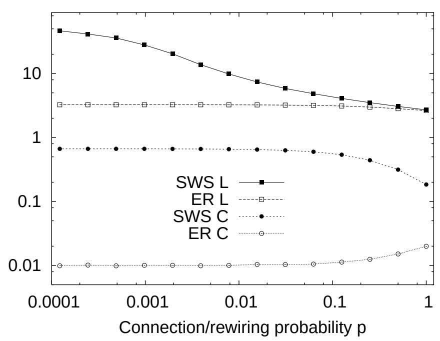


FIGURE 5.1: On the left, $\mathcal{C}(G)$ and $\mathcal{L}(G)$ normalized by $\mathcal{C}(C_{n,k})$ and $\mathcal{L}(C_{n,k})$ respectively for $p \in (0, 1]$. Sample graphs are generated for parameter values $p = 1$, 0.5, 0.25, 0.125, ... until $p < 0.0001$. The values are averages over at 30 random realizations of the SWS model for $n = 1{,}000$ and $k = 5$. Data of Figure 3.4 for the WS model from [134] is included for comparison. On the right, the unscaled values of $\mathcal{C}(G)$ and $\mathcal{L}(G)$ averaged over a set of 30 random realizations are shown together with the respective values for $\mathcal{G}_{n,m}$ instances generated to match the order and size of the 30 random samples for each value of p.

the case of $p = 1$ differs as for the original model because the WS graphs will be more random as all edges are rewired, whereas SWS maintains the clustering inherent in the underlying $C_{n,k}$. Measurements for both the original WS model and our implementation of the SWS model are shown in Figure 5.1.

5.1.3 Undirected Kleinberg lattice model

As Kleinberg's model (described in Section 3.2.2) is a directed model, unlike any other used in the experiment set, the implementation ignores the directed nature of the randomly added links that reach outside the local neighborhood to make these graphs directly comparable to those of other generators. The vertices of the lattice are labeled by their Euclidean coordinates on the $s \times s$ square lattice. The coordinate labels (x, y) are mapped to single integer values ℓ such that the upper left corner of the lattice is assigned label zero, the one below that vertex will be labeled as number one, and so forth until the bottom of the lattice is reached. Then the labeling continues from the top vertex of the next "column" to the right. This simple "top-down left-right" labeling allows also a simple mapping to the coordinate labels: $x = \lfloor \frac{\ell}{s} \rfloor$, $y = \ell \mod s$. The lattice distance calculation is therefore simple to implement by taking advantage of C's integer arithmetic operations.

The *local neighborhood* by definition contains all vertices that can be reached by taking at most p "steps" on the lattice, that is, with Manhattan distance $dist_{\mathbb{L}} \leq p$ (see Equation 3.15 on page 37). Therefore the size of the local neighborhood of "radius" $p \geq 1$ is at most $2p(p + 1)$, as observed from Figure 5.2. If the distance

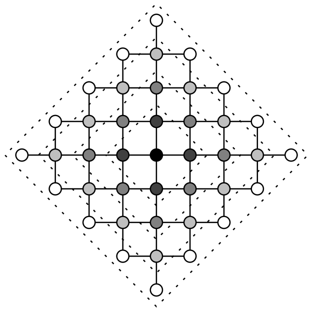


FIGURE 5.2: The p-neighborhoods of a single vertex v (drawn in black) in a Kleinberg lattice graph for $p \in \{1, 2, 3, 4\}$. Each p-neighborhood is encompassed by a dotted square and the newly reachable vertices have lighter color than those of the smaller neighborhood. The grid is drawn for interpreting the lattice distances; it is not the edge set of the graph.

of a vertex is more than p from the lattice border, this will be exact, otherwise an upper bound. In addition to the local p-neighborhood, each vertex is linked to q vertices that are further away than p steps. The linking probability is proportional to the negative rth power of the lattice distance. To add q long-distance neighbors for vertex v, we first count the normalizing sum for v, namely

$$S = \sum_{u \notin \Gamma(v) \cup \{v\}} \frac{1}{dist_{\mathbb{L}}(u, v)^r}, \tag{5.5}$$

then obtain a uniformly distributed random value $\rho \in (0, 1)$, multiply by S to obtain the "target value" $t = S\rho$ and then build an incremental sum s_i of the values $1/(dist_{\mathbb{L}}(u, v))^\rho$ for all $u \notin \Gamma(v) \cup \{v\}$ until $s_i \geq t$. For the vertex w for which this first happens, we add the edge (v, w) to E. If $q > 1$, we draw another random value ρ' and repeat the process until either v is linked to all other vertices or q long-range connections have been formed.

The limiting distance p is a parameter of the model, required to be an integer greater or equal to one. In general, p is quite small as otherwise it would be a poor parameter of locality. The graph is expected to be sparse for $p \ll s$. The q random edges do not significantly increase the degree of any particular vertex, as they are connected further away from the source vertex and the distances are Euclidean grid-distances. The best presentation form would be an adjacency list instead of an adjacency matrix. However the generation algorithm becomes somewhat messy to implement with lists as the neighborhood relation is constantly browsed, and hence we chose to use the adjacency matrix. The parameter r is fixed to two (as Kleinberg [78] recommends) for the experiments, although a parameter of the generation procedure.

We compared the clustering coefficient $\mathcal{C}$ and the characteristic path length $\mathcal{L}$ of the KL model to those of $\mathcal{G}_{n,m}$ graphs of the same order and size, using different values of p and q for $s = 25$ and $r = 2$. As the calculation of the pairwise distances is computationally demanding, the order of the graphs was kept relatively small. The results are shown in Table 5.1.3.

TABLE 5.1: The values of clustering coefficient $\mathcal{C}$ and characteristic path length $\mathcal{L}$ for some KL graphs with $s = 25$, $r = 2$, and several values of p and q. Also the corresponding values for $\mathcal{G}_{n,m}$ graphs with same order and size are shown to ease comparison. Each cell of the below tables contains the four values in the below order, all being averages over at least 30 independent instances.

$\mathcal{C}_{\mathrm{KL}}$	$\mathcal{L}_{\mathrm{KL}}$
$\mathcal{C}_{\mathrm{rand}}$	$\mathcal{L}_{\mathrm{rand}}$

$s = 25 \Rightarrow n = 625$

q \ p	1		2		4		8	
0	0.000	16.67	0.475	8.584	0.574	4.543	0.641	2.523
	0.007	4.878	0.018	2.913	0.057	2.071	0.181	1.819
1	0.083	4.673	0.378	3.590	0.531	2.726	0.627	2.095
	0.009	3.841	0.021	2.762	0.060	2.041	0.184	1.816
2	0.103	3.789	0.318	3.154	0.494	2.543	0.614	1.982
	0.013	3.357	0.024	2.661	0.063	2.016	0.187	1.813
4	0.115	3.120	0.250	2.762	0.438	2.361	0.591	1.877
	0.019	2.859	0.031	2.506	0.069	1.977	0.194	1.807
8	0.132	2.619	0.204	2.451	0.370	2.143	0.553	1.809
	0.032	2.484	0.044	2.249	0.082	1.931	0.206	1.794

Graphs that meet the small-world requirements of Watts and Strogatz have $\mathcal{L} \approx \mathcal{L}_{\mathrm{rand}}$ and $\mathcal{C} \gg \mathcal{C}_{\mathrm{rand}}$. In Table 5.1.3, the small-world phenomenon is the most evident for graphs with $q \in \{1, 2\}$. In general, the value of q needed to produce the drop in $\mathcal{L}$ without significantly disturbing $\mathcal{C}$ is small in comparison to the size of the p-neighborhood. Clustering properties of KL graphs for $s = 100$ and eleven different (p, q)-pairs are listed in Section 5.2. We also plotted in Figure 5.3 the degree distributions of some KL graphs for comparison with the other models. The small jumps in the distributions are caused by the boundary conditions of the lattice: the closer to the end of the lattice a vertex is, the more of its p-neighbors are absent from the graph.

5.1.4 Barabási-Albert model with tunable clustering

For the BA model, described in Section 3.3.1, the initial graph is chosen to be a connected random graph with n_0 vertices, as using an empty graph would initially cause division by zero in Equation 3.25 of the preferential attachment probabilities. The initial graph must be connected to ensure that a connected graph will result regardless of the random linking. We iterate the $\mathcal{G}_{n_0,p'}$ construction starting with $p' = \frac{2}{n_0}$ and increasing p' in steps of 0.05 until a connected sample is obtained.

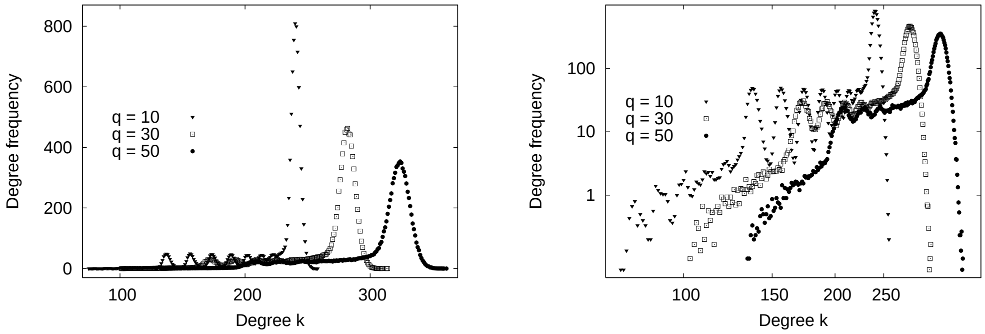


FIGURE 5.3: On the left, degree distributions of some KL graphs, shown on log-log scale on the right. The curves are averaged over 30 independent instances. The generation parameters are $s = 100$, $r = 2$, $p = 10$, and $q \in \{10, 30, 50\}$.

Note that the graph could not be connected if it had less than $n_0 - 1$ edges, and also that $|E_0| = m_0$ is not constant for a given n_0 in this implementation.

At time step t, a new vertex v_t is introduced and assigned d distinct edges that link it to the graph G_{t-1} of the previous step. It is clear that $n_0 \geq d$ must hold for the first step to be well-defined. The preferential attachment is implemented by retrieving a uniformly distributed random integer r from the range $[0, \sum_v \deg(v))$ and then incrementing a counter c by adding the degree values $\deg(v_0)$, $\deg(v_1)$, $\ldots$, $\deg(v_{t-1})$ one at a time. When the counter value c first exceeds the random integer r after adding $\deg(v_k)$, the vertex v_k is chosen as the target vertex of the preferentially attached edge. This is repeated until such vertex v_k is found for which $(v_t, v_k) \notin E_t$, after which the edge is included in the graph. Such preferential attachment is iterated until d distinct edges have been placed, which necessarily happens as $n_0 \geq d$ and $n_t = n_{t-1} + 1$. The degree of the vertex v_k and hence also the sum of degrees will not be incremented until all d edges have been assigned, in order to maintain the preferential distribution the same for throughout the time step t. Before the time $t + 1$, the degree of v_t is set to d and the degrees of all its new neighbors incremented by one. Note that $n_{t+1} = n_t + 1$ and $m_{t+1} = m_t + d$.

The form of the distribution in Figure 5.4, containing thirty independent BA instances, is very similar to that of Barabási and Albert [12]. The figure suggests that the graphs generated by this implementation are scale-free in the sense used in recent literature on nonuniform networks: the distribution falls near a straight line on a log-log plot as expected. The usefulness of such diagrams seems controversial, but much of the discussion on scale-free topologies relies on determining the exponent γ for different models and network instances. We therefore follow this convention and determine γ for our implementation. Summing over the thirty independent BA instances and fitting a line $f(x) = \gamma x + c$ on the logarithms of the

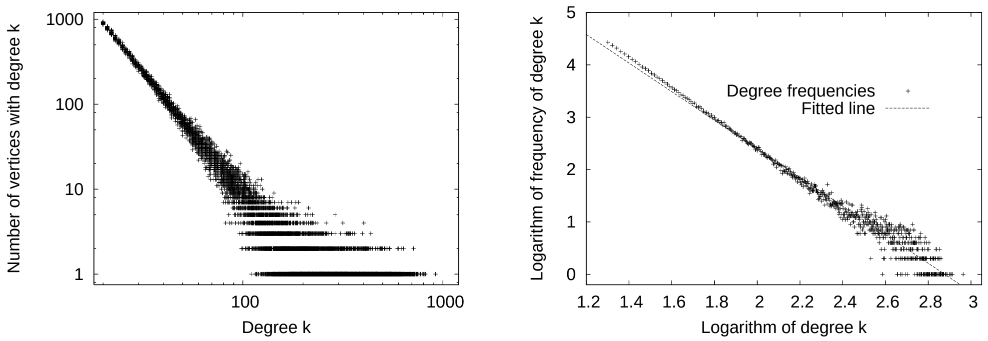


FIGURE 5.4: On the left, a plot of the degree distribution of thirty independent BA graphs with parameters $n_0 = d = 20$, $n = 10{,}000$. For each degree k present, the number of vertices with that degree is shown. The right side shows a plot obtained by averaging over the degree frequencies of the thirty instances shown on the left.

data with `gnuplot` yields $f(x) = -2.7317x + 7.85957$. With asymptotic error, we have $\gamma = -2.7317 \pm 0.0192$. This total distribution and the fitted line are shown on the right Figure 5.4. By limiting the fitting to $x \in [1.2, 2.4]$ we obtain line that visually judging follows the shape of the distribution better, as the "noisy" spread at the low-frequency degrees is eliminated. The line fitted only on the limited interval is $g(x) = -2.92712x + 8.24662$, with γ closer to the analytical result of three.

The implemented model includes a clustering step of Holme and Kim [65] (see end of Section 3.3.1) and is therefore abbreviated as the CBA model. The probability $p_{\mathcal{C}}$ that a clustering step will follow the first preferential linking is a parameter of the generation procedure. If the probability parameter is given value zero, the step is omitted and all d links are attached preferentially. At time t, after the first of the d edges has been assigned preferentially as (v_t, u), a uniformly distributed number $r \in [0, 1]$ is drawn. If $r \leq p_{\mathcal{C}}$, the triangle formation step is attempted. A uniform random integer $r' \in [0, n)$ for a starting point of a wrapping search of a neighbor of u that is not yet a neighbor of v_t. As such a vertex is encountered, triangle formation takes place. If all $n - 1$ vertices are improper for triangle formation, we instead perform preferential linking. After a successful triangle formation, we test by drawing a new r whether to perform another triangle formation step or return to preferential linking. This is repeated until $\deg(v_t) = d$.

The generated topologies remain scale-free when the clustering step is applied, a claim supported by our experimental data shown on the right in Figure 5.5, similar in shape to that of the BA model shown in Figure 5.4. Also the experiments of Holme and Kim produce similar distributions. Fitting lines to the distributions of the figure, we obtain values of γ varying from 2.00311 ± 0.05274 for $p = 0.0$ to as low as 1.30135 ± 0.06415 for $p = 1.0$ when fitting to the entire distribution. When

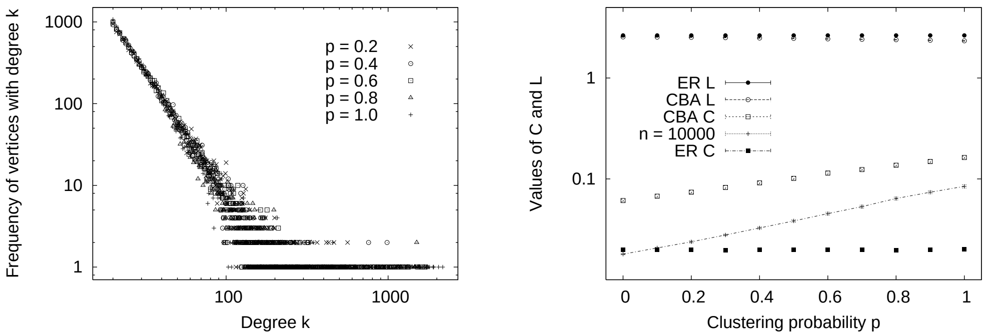


FIGURE 5.5: On the left, degree distributions of five CBA graphs with n = 10,000, $n_0 = d = 20$, and varying clustering probabilities p. On the right, plots of the clustering coefficient $\mathcal{C}$ of CBA graphs with n = 1,000, $n_0 = d = 10$ together with $\mathcal{G}_{n,p}$ graphs (the ER model) generated to match the order and size of the CBA graphs. Also, values of $\mathcal{C}$ for CBA graphs with n = 10,000, $n_0 = d = 20$ are shown. The values are averages of at least 30 graphs, standard deviation shown.

the fitting was limited to range $x \in [1.5, 2]$ to eliminate the noise at the end, the values ranged from 2.98301 ± 0.08173 for $p = 0.0$ to 3.30847 ± 0.1155 for $p = 1.0$.

We ran some tests to examine the increase in clustering as $p_{\mathcal{C}}$ is increased, which turns out to be rather small for many parameter sets. It can easily be seen from Figure 5.5 that the CBA graphs are not small-world graphs in the same sense than the WS or SWS graphs (see Figures 3.4 and 5.1); although $\mathcal{L} \approx \mathcal{L}_r$, also $\mathcal{C} \approx \mathcal{C}_r$.

5.1.5 Deterministic clustered scale-free model

The clustered deterministic RB model, also described in Section 3.5, starts with an initial complete graph K_{n_0}. Our generation procedure takes the order of the initial graph as a parameter, but we have restricted our experiments to K_5 as originally defined by Ravasz and Barabási [117]. One of the vertices of the initial graph is chosen as the *root* vertex, the others are marked as "peripheral" vertices. A new generation G_t is created by taking four copies of the previous graph G_{t-1}. Also the copy count is defined as a parameter to the implemented procedure although it is in the experiments fixed to four as in [117]. In the resulting disconnected graph, all vertices that are a copy of a peripheral vertex are marked peripheral, and the peripherality mark is removed from previously peripheral. All new peripheral vertices are connected by an edge to the root vertex, which joins copies of G_{t-1} are joined with G_{t-1} to form a connected G_t.

Our interpretation of the construction is based on personal communication with Erzsébet Ravasz, as the formulas for $|V_t|$ or $|E_t|$ are not given in [117] and the

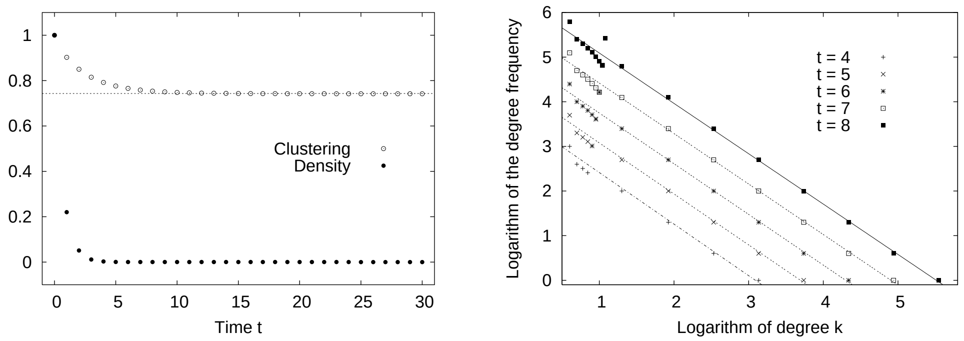


FIGURE 5.6: On the left, density δ and clustering coefficient $\mathcal{C}$ of some G_t generations; the horizontal line shows the limit value 0.743 obtained by Ravasz and Barabási in [117]. On the right, degree distributions of some G_t generations with lines fitted to the plots. Ravasz and Barabási predict $\gamma \approx 2.161$.

description of the model is quite succinct. Ravasz and Barabási [117] report numerical simulations indicating $\mathcal{C} \approx 0.743$, whereas we obtained $\mathcal{C} \approx 0.74184$. Some of the first values of $\mathcal{C}(G_t)$ together with $\delta(G_t)$ are shown on the left in Figure 5.6.

Ravasz and Barabási also find these graphs scale free with $\gamma = 1 + \frac{\ln 5}{\ln 4} \approx 2.161$. A plot of the degree distribution of G_8 (n = 1,953,125 and m = 9,107,674) together with some earlier generation is given in Figure 5.6. Fitting lines to these distributions with `gnuplot` we note that the slope seems to be constant. The slopes of the lines fitted to the distributions are given in Table 5.2 and average at $\gamma \approx 1.138$, close to $\frac{\ln 5}{\ln 4} \approx 1.161$. In personal communication, Ravasz explained the value of γ being higher than this due to the gaps in the degree distribution; not all values of k are present in the graphs at time t, but only a restricted subset.

TABLE 5.2: The slopes $-\gamma_t \pm \epsilon_t$ of the lines (with asymptotic standard error ϵ_t) fitted with `gnuplot` to the distributions of the respective G_t in the degree distribution plots of Figure 5.6.

t	$-\gamma_t$	ϵ_t
4	-1.1511	± 0.08656
5	-1.1426	± 0.05714
6	-1.13569	± 0.04035
7	-1.1313	± 0.03015
8	-1.12819	± 0.02373

5.1.6 Hierarchical caveman model

The connection probability $p \in (0, 1]$ of the top level of the hierarchy is given as a parameter, together with a scaling coefficient s that adjusts the density of the lower-level caves. The minimum $n_{\min}$ and maximum $n_{\max}$ for the numbers of subcomponents (subcaves at higher levels, vertices at the bottom level) are given as parameters. The generation procedure is recursive; a brief description is given below and a sample graph is shown in Figure 5.7. These graphs all have high clustering and relatively short path length by construction unless both the initial connection probability and the scaling factor are set to produce sparse caves and a sparse hierarchy. The example graph of Figure 5.7 has $\mathcal{C} \approx 0.82$ and $\mathcal{L} \approx 2.51$, whereas the respective values for ER graphs are $\mathcal{C}_{\text{rand}} \approx 0.15$ and $\mathcal{L}_{\text{rand}} \approx 2.13$, averaged over a set of at least 30 instances.

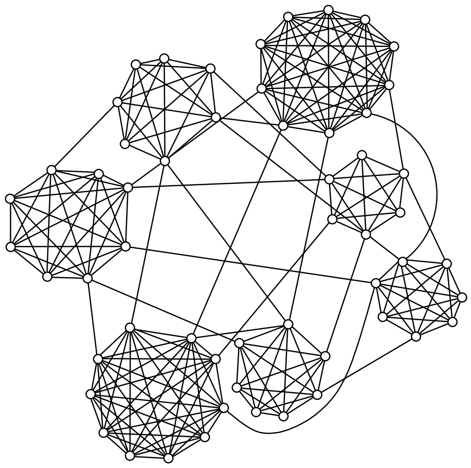


FIGURE 5.7: An example graph, $n = 55$, $m = 217$, generated with the hierarchical caveman model that has seven caves. The generation parameters were $\ell = 1$, $n_{\min} = 5$, $n_{\max} = 10$, $p = 0.8$, and $s = 1.2$. If the other parameters are kept fixed but $\ell = 2$ and $p = \frac{2}{3}$, the top-level caves all resemble this graph.

A cave at a certain level ℓ of the hierarchy is formed of a random number $r \in [n_{\min}, n_{\max}]$ of subcaves with connection probability sp', where p' is the connection probability at level ℓ. If $sp' \geq 1$, the connection probability of the next and lower levels will be one. Each subcave is either a hierarchical cave, or at the bottom level, a random graph of the $\mathcal{G}_{n,p_b}$ with a random $n \in [n_{\min}, n_{\max}]$; p_b is the connection probability of the bottom level. A cave that consist of subcaves is randomly connected into a larger graph; the connections are placed as in a $\mathcal{G}_{n,p}$, considering the subcaves as single vertices, the inter-cave connection being assigned to a random member at each subcave.

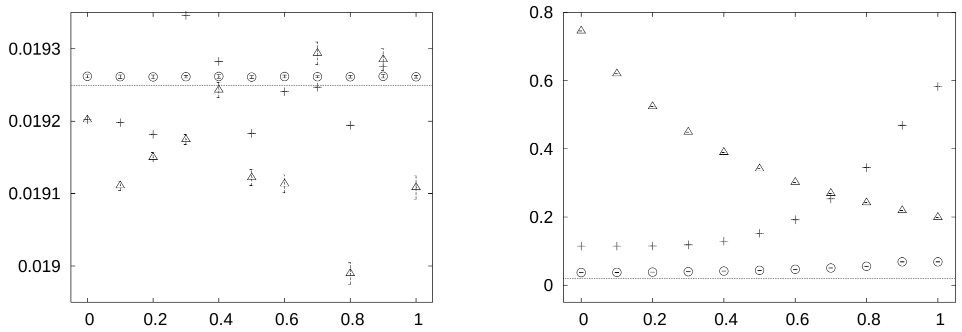


FIGURE 5.8: Average values of density δ (left) and clustering coefficient $\mathcal{C}$ (right), plotted for the CBA ($\circ$), SWS ($\triangle$), and KL ($+$) test sets. The x-axis shows the clustering probability p for CBA, rewiring probability p for SWS, and $\frac{p}{10}$ for the local neighborhood radius p of the KL model. The vertical lines represent the average values over 31 independent $\mathcal{G}_{n,d}$ instances with $n = 10{,}000$ and $d = 0.01925$.

5.2 Algorithmic implications

We have studied the behavior of an algorithm for the MAXIMUM CLIQUE problem by Patric Östergård [122], provided in the `cliquer` library implemented by Sampo Niskanen [108], available online at `http://www.hut.fi/~pat/cliquer.html`. We measured the running time of the `clique_unweighted_max_weight` routine with random vertex labeling. We used just one workstation and hence had reasonable control over the load during the test runs. For more extensive tests, the number of elementary operations of interest should be counted when feasible, as running time is not a good measure of algorithmic performance under varying computer load and details of the hardware configurations as well as program optimization [73].

TABLE 5.3: The parameters of the `cliquer` tests.

Model	Parameter p		Studied	Step	Model parameters
CBA	clustering step probability	$p \in [0,1]$	$[0,1]$	0.1	$n_0 = 500$, $d = 100$
SWS	rewiring probability	$p \in [0,1]$	$[0,1]$	0.1	k from δ
KL	local neighborhood radius	$p \in \mathbb{N}$	$[0,10]$	1	$s = 100$, q from δ

To examine whether the clustering of a graph affects the performance of the algorithm of the MAXIMUM CLIQUE problem provided in the `cliquer` library, we generated test sets with CBA, SWS, and KL models. All graphs are of order $n = 10{,}000$ and have density $\delta \approx 0.019$. Averages for density and clustering coefficient of the graphs are shown in Figure 5.8. The density of the CBA test is $\delta \approx 0.01925$, which was matched as closely as possible for the other two models by properly fixing the other model parameters, using the definition of density $\delta = m/\binom{n}{2}$. For

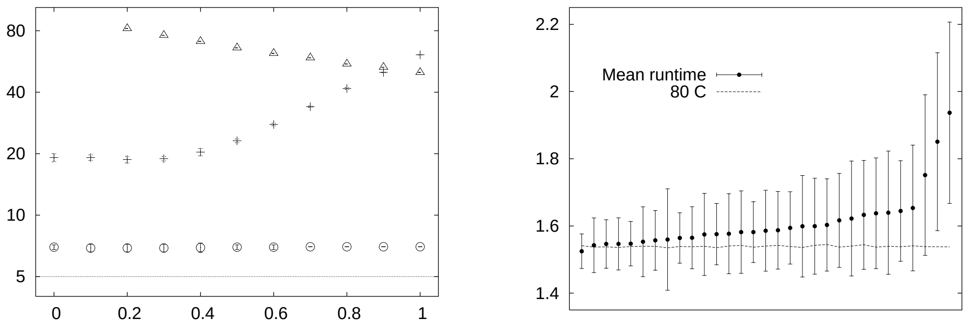


FIGURE 5.9: On the left, average values of maximum clique order plotted for the CBA (○), SWS (△), and KL (+) test sets. The x-axis shows the clustering probability p for CBA, rewiring probability p for SWS, and $\frac{p}{10}$ for the local neighborhood radius p of the KL model. All ER instances had cliques of 5 vertices. On the right, the average running times of the independent ER instances sorted in increasing order, averaging at 1.6 seconds. The dotted line shows the clustering coefficient scaled by 80 per each instance.

SWS, the regular connection distance used is $k = (\delta(n-1))/(2(1+p))$ rounded to the closest integer value, using the expected number of edges $\mathrm{E}[m]$. For KL, the number of long-distance connections q is derived from the exact edge count m, defined in Section 3.2.2. For comparison, ER graphs were generated from the $\mathcal{G}_{n,p}$ family with the same order and connection probability $p = 0.01925$.

A relatively large order was chosen, as problems resulting from small graphs have been observed in the experiments of [17], where the asymptotic region of the measured properties was not reached. The varied parameters are shown in Table 5.3. We generated 31 instances for each parameter value, obtaining in total 341 graphs per test set. We ran `cliquer` at least 30 times on each graph to measure the variations of the running time. For the ER graphs, the average running times over the set of independent instances, all having the same generation parameters, are shown in Figure 5.9.

The running times for the CBA (top), SWS (middle), and KL (bottom) models vary as shown in Figure 5.10. For some of the parameter values, the running times of `cliquer` are very high and hence examining all 31 instances with at least 30 repetitions is infeasible. Therefore `cliquer` was only ran once per instance for $p \in [0.4, 0.7]$ for the SWS model and $p > 8$ for the KL model, which decreases the reliability of these runs. For the SWS model, the instances for which $p \in \{0.0, 0.1\}$, examining even a single instance is so slow that `cliquer` takes more than three days to complete and hence these runs are skipped. For $p \in \{0.2, 0.3\}$ that are also very slow, we ran `cliquer` only once for just one instance to demonstrate the increase in running times.

The running time clearly grows with clustering coefficient, but much faster. The

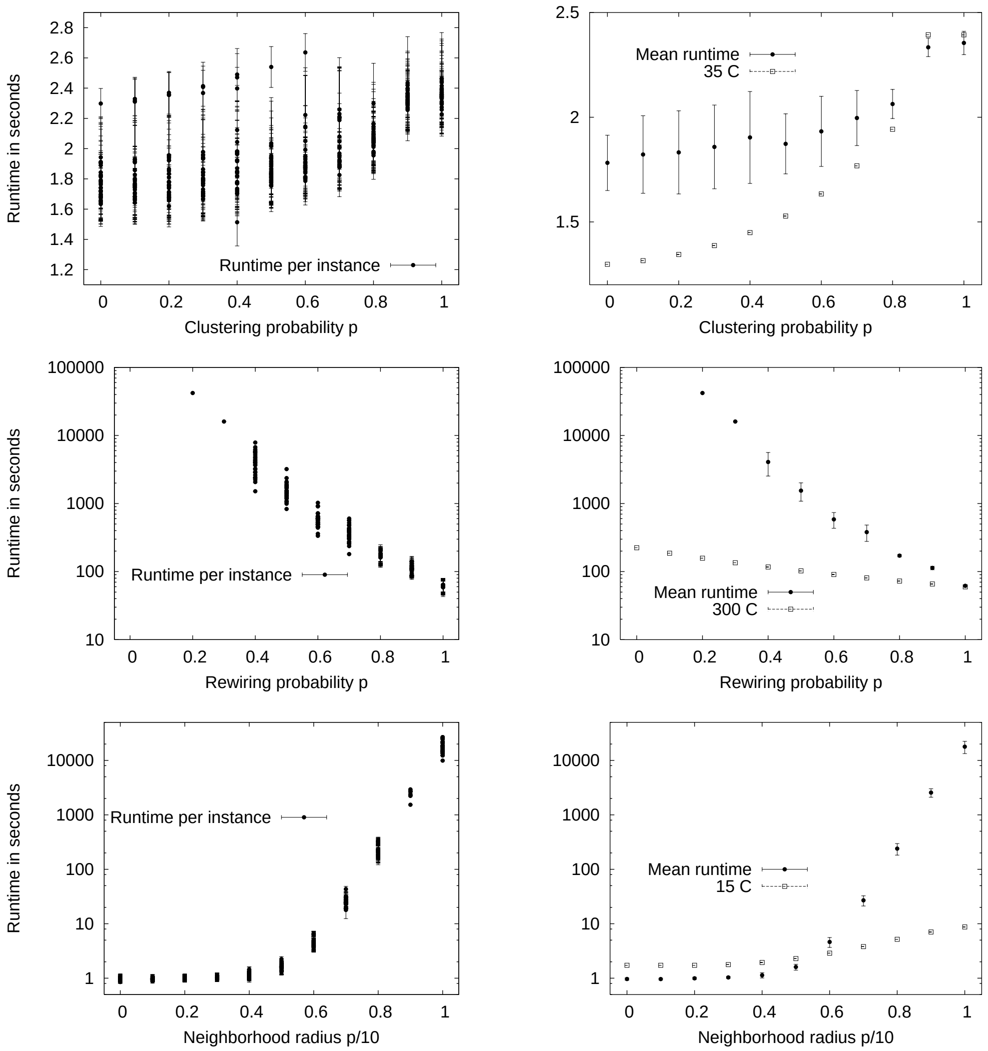


FIGURE 5.10: Running times of `cliquer` for the CBA (top), SWS (middle), and KL (bottom). On the left, the average running time for each instance is shown. On the right, the mean values of the average running time of `cliquer` together with a scaled plot of $\mathcal{C}$ are shown. In all plots, standard deviations are drawn.

highly clustered graphs of the SWS model are almost all difficult instances for cliquer. The clique order is plotted in Figure 5.9. Graphs with larger cliques take more time to examine, and the clustering coefficient, as a measure of "cliquishness", measures how likely is it for a neighborhood of a vertex to be a clique. However, as Figure 5.9 shows for the ER running times, changes in $\mathcal{C}$ alone do not determine the running time; as $\mathcal{C}$ approaches one half, the running time of `cliquer` rapidly rises from just a couple of seconds (CBA, ER, and KL for small values of p) to several minutes (KL for large values of p, SWS). As all of these graphs have the same order and similar size (as their densities have been adjusted to match as closely as possible), it is apparent that neither order nor density are sufficient predictors of the running time. We are interested to study this phenomenon further; for example by studying the distribution of clique orders in the graphs.

5.3 Properties of natural graphs

In addition to graphs produced by generation models, we found it informative to study graphs that have been formed from real-world data. We call such graphs *natural* to make the distinction to those artificially constructed by a generation model. One example of natural graphs is the neural network of the *C. elegans* presented in Section 2.1, for which values of $\mathcal{C}$ and $\mathcal{L}$ were shown already in Table 3.1 on page 36. Farkas et al. [47] use spectral properties to classify small networks that are constructed on real data. They compare the spectrum of a graph to the spectral properties known for different families (such as ER, WS and BA models of Chapter 3) and interpret which one fits the given data the best. We do not resort to spectral methods in this study.

The DIMACS[3] benchmark graphs for coloring and clique problems[4] are one collection of such graphs originating from different applications. We studied the ASCII format graphs (`*.col`). Any duplicate edges present were ignored and all graphs were treated as undirected. Disconnected graphs were also ignored, as well as the Mycielski transformation graphs, which are triangle-free and therefore have zero clustering by definition.

The SGB road mileage graphs of the DIMACS benchmark set have a set of 128 U.S. cities as vertices and an edge (u, v) if the road mileage from the city represented by u to that represented by v is *smaller* than a threshold. For a threshold of 250 miles the graph is disconnected, but the other SGB road mileage graphs are connected and hence have well-defined average path length. The values of $\mathcal{L}$ and $\mathcal{C}$ for some of these graphs are shown on the left in Figure 5.11 together with the respective data on random graphs. The graphs display some small-world behavior, as the clustering coefficient is considerably higher and the average path length is of similar magnitude for all of the road mileage graphs than their random counterparts. Hence the small-world phenomenon is present. As the connection distance grows, the graphs seem to behave more similarly.

The SGB Queen graphs are representations of dependencies between the squares of a $N \times N$ chess board when solving the N queens problem. If such a graph is N-colorable, then the N queens problem has a solution. We studied the clustering and path length properties of twelve of these graphs, the results shown on the right in

[3] `http://dimacs.rutgers.edu/`

[4] `http://mat.gsia.cmu.edu/COLOR/instances.html`

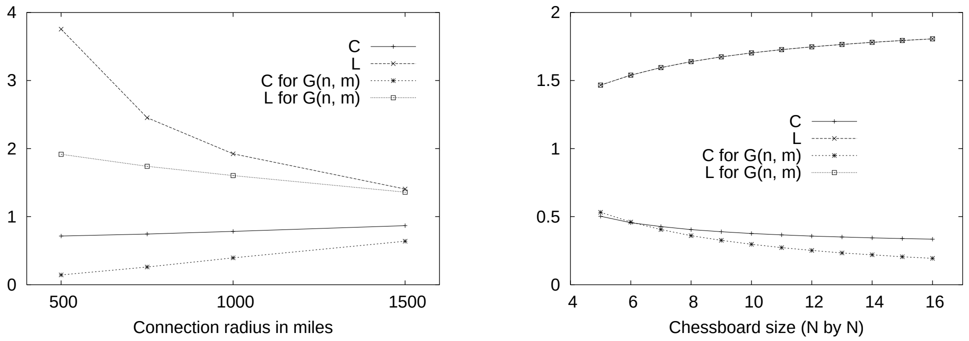


FIGURE 5.11: The clustering and path length behavior of some SGB graph compared to $\mathcal{G}_{n,m}$ graphs of the same n and m. The values are averages over 30 $G_{n,m}$ graphs. On the left are road mileage graphs for different connection distances and on the right dependency graphs of N-queens puzzles for $N \in [5, 16]$.

Figure 5.11. Note how the path length is almost exactly the same as for a random graph of the same parameters but clustering decays slower for the Queen graphs than for the random graphs. Hence the small-world effect cannot be observed for these graphs.

Newman [101] has studied scientific collaboration networks constructed from other databases, such as the Los Alamos e-Print Archive. He argues that the coauthorship in scientific publications is closer to true social acquaintance than the IMDb network of Section 2.1 and believes his reconstruction of the collaboration network from database entries to be the first of its kind. Newman has chosen to identity authors by two alternative definitions: either by using the last name and the first initial or by using all of the initials for each author. He believes the former to provide a lower bound and the latter an upper bound to the number of truly separate authors, but as bibliographic data is often quite varying in quality, this claim is probably an educated guess rather than firm knowledge. The former method may cause several authors to be considered a single person, whereas the latter introduces the chance that a single authors “splits” into two vertices in the network as different publication fora commonly use different numbers of initials.

We downloaded several bibliographies from The Collection of Computer Science Bibliographies [2] and built a collaboration graph based on this data. To limit the order of the resulting network, we downloaded only the mathematical bibliographies as listed at `http://liinwww.ira.uka.de/bibliography/Math/` on December 2, 2002. We retrieved only those bibliographies that were available in BibTeX format. The sample includes 379 files from the FTP server of the Department of Mathematics at the University of Utah[5] and about 50 other files accessible through [2]. Only eight bibliographies were unavailable at the time.

[5] Available online at `ftp://ftp.math.utah.edu/pub/tex/bib`.

The files were processed with a simple Java-program that parses the BibTeX format, trying to ignore authors that are not persons (such as institutes and committees), simplifying the spelling of the names, ignoring Roman numerals, and interpreting which word is the first name and which is the last name of an author. All BibTeX-fields that are not `author`-fields are ignored entirely, as well as comments. As the bibliographic data was somewhat diverse and especially all exotic names have varying forms of spelling even within just one bibliography file, we represent all authors with the same first initial and surname by the same vertex. For comparison we also tested a construction in which only the surname was used. Dashes and other such characters in the names were removed, and special Unicode characters were replaced by their ASCII counterparts. Even with the above simplifications, more than 170,000 bibliographic entries with multiple authors were found. Each such entry is represented by a line in the parser output that defines the "vertex labels", which are the simplified last names of the authors, with duplicates eliminated. For example,

```
< ecateland gskordev hpeitgen jallouche jshallit wgilbert >
```

This data was translated to simplified DIMACS graph format with another Java-program. The first line of the output is "`p edge` n m", where n and m define the number of vertices and edges respectively. Each edge (v, w) is represented by a line "`e` v w". Multiple and reflexive edges were omitted. For more details of the parsing and the related simplifications, see the source code. An example of a collaboration graph is given in Figure 5.12, where the bibliographic entries of this work have been parsed into a graph; the figure only shows the largest connected component.

The graph that results from joining all the above BibTeX files with just the surname as author identification has 78,758 vertices and 331,551 edges. Adding the first initials increases it to 129,215 vertices and 350,914 edges. We denote the former graph by G_{last} and the latter by G_{init}. In the largest database used by Newman [101], the MEDLINE database for biomedical research, there were 1,090,584 vertices when authors were classified by first initial and last name and as many as 1,520,251 when classified by all initials — the true number of authors is likely to be somewhere in between. This difference is of similar magnitude than the difference resulting in our greater simplification with last names only versus first initials and last names. In the smaller databases studied by Newman, the relative difference was somewhat smaller.

Our collaboration graphs are both very sparse; $\delta(G_{\text{last}}) \approx 6 \cdot 10^{-4}$ and $\delta(G_{\text{init}}) \approx 4 \cdot 10^{-5}$. The largest connected component of G_{last} has 73,707 vertices and 327,891 edges, therefore covering 93.6 % of the network. For G_{init}, the connected component covers 84.1 percent of the graph with 108,624 vertices and 333,546 edges. In our experiments we concentrate on the connected subgraph of G_{init}, obtained by yet another Java-program. It is noteworthy that as these graphs are large and the Java

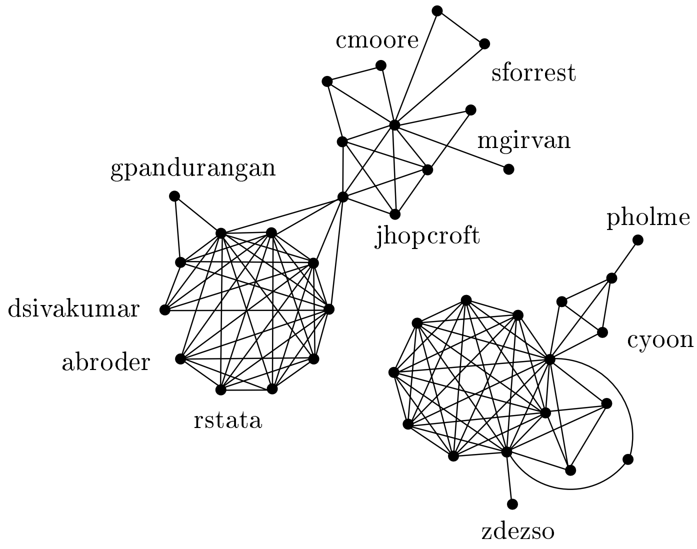


FIGURE 5.12: The two largest connected components of a collaboration graph based on the BibTeX file of this work. Only some vertices are labeled to keep the picture clear.

programs are all but optimized, some of the computations can be quite lengthy and were better left to run overnight. We will consider integrating these tools to the C library of graph generation and analysis as further work, possibly introducing some optimization as well.

We concentrate on the largest connected component of G_{init}, denoted by G_c with $n = 108{,}624$ and $m = 333{,}546$. The order of the second largest component is significantly smaller; it contains only 20 vertices. In total there were 7,838 components in G_{init}. Excluding G_c, the average order of these components is only 2.6 and the median order 2. We calculated some of the basic measures for G_c, obtaining density $\delta(G_c) \approx 5.65 \cdot 10^{-5}$, average degree $\overline{k} \approx 6.14$ and girth $g = 3$. The clustering is quite high with $\mathcal{C} \approx 0.64$. The diameter of the network is as high as 22 and the average path length is $\mathcal{L} \approx 5.94$ — almost exactly "six degrees of separation" between two authors. As $\mathcal{L}(G_c)$ is considerably smaller than $\text{diam}(G_c)$ and the clustering is fairly high, it is safe to say that the collaboration graph exhibits the small-world property.

The degree distribution of the connected collaboration graph is shown in Figure 5.13; it somewhat resembles the scale-free distribution of the BA model. Fitting a line with `gnuplot` to the log-log plot of the degree distribution yields $\gamma = 2.40746 \pm 0.04618$; the line $f(x) = -\gamma x + b$, where $b = 5.56419 \pm 0.08804$ is shown in Figure 5.13. Note that if the degrees with frequency one are ignored, the fitted line appears to match closer the slope and position of the distribution. The slope of the obtained line is 2.39494 ± 0.04694.

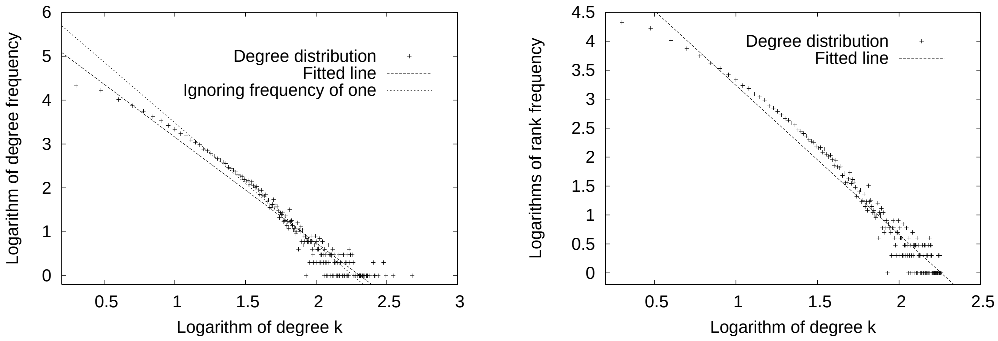


FIGURE 5.13: Degree distribution of the collaboration graph as a log-log plot on the left, and on the right, the degree rank distribution of the same graph.

Newman [101] finds that his collaboration graphs do not perfectly follow a power-law, but rather a from with *exponential cutoff*

$$P(k) \sim k^{-\gamma} e^{-k/c}, \tag{5.6}$$

where c is a constant. He finds $\gamma \approx 2.5$ and $c \approx 5{,}800$ for his MEDLINE collaboration graph (see Section 5.3). It would be interesting to find better fitting curves for our data as well when continuing work on this area, as there is apparent curvature on both ends of the log-log plot. A debated issue with plots of degree distributions is whether to examine them as a power-law using the degree values or to examine them as a Zipfian distributions (see page 19) using the ranks of the degrees. The "noise" on the bottom fades nicely when plotting the ranks of the degrees instead of the degrees themselves also for our collaboration graph. This is shown on the right in Figure 5.13; the fitted line is $\gamma = 2.57468 \pm 0.04372$ and $b = 5.81331 \pm 0.08208$.

Some of the vertices have surprisingly high degree as the number of collaboration partners is unlikely several hundreds. We traced the names corresponding to then ten vertices for which $\deg(v) > 250$. Nine of these were to Asian surnames with a common first initial: `jlee` has the highest degree, namely 478, and `slee` has degree 351. As also `dlee` is among the ten highest degrees with 254 neighbors, it is obvious that using the last name only would combine all these to a vertex with disturbingly large degree: vertex `lee` in G_{last} has degree 1,586, which is only exceeded by `smith` with 1,679 neighbors. Barabási et al. [14] have noticed the same problem with surnames of Chinese and Japanese descent. The only English surname appearing in the top ten is `dcrawford` with degree 275, presumably largely attributed to a single author. In comparison, Pál Erdős, being a very productive scientist, has published papers in collaboration with some 500 authors. Choosing to include the first initial makes the network much more realistic considering the above observations. Yet even then we have reason to believe that most of the high-degree vertices correspond to more than one author and can therefore be considered noisy data that distorts the shape of the degree distribution.

TABLE 5.4: Some measurements on the collaboration networks derived from the Computer Science Bibliographies archive's mathematical articles (CSB) and the publications stored in the Los Alamos e-Print Archive (LAE). The latter data is from [101]. Note that the diameter is defined for the largest connected component. The CSB measures for the average degree, distance, and clustering are calculated for the largest connected component, whereas we presume Newman's calculations to include the entire graph.

Measure	CSB	LAE
Total order $\|V\|$	129,215	52,090
Order of the largest component	108,624	44,336
Percentage covered by the largest component	84.1	85.4
Order of the second largest component	20	18
Average number of collaborators $\overline{k}$	6.14	9.7
Average shortest distance $\mathcal{L}$	5.94	5.9
Diameter	22	20
Clustering coefficient $\mathcal{C}$	0.64	0.43

For comparison, we briefly summarize some of Newman's results from [101]. For his coauthorship graphs of scientific publications, diam $\in [14, 31]$. For the graph containing complete Los Alamos e-Print Archive data, which is of similar order than ours with 98,502 vertices, the diameter is 20 and $\mathcal{L} \approx 5.9$, whereas we obtained $\text{diam}(G_c) = 22$ and $\mathcal{L}(G_c) \approx 5.94$ for our graph. For Newman's graphs $\mathcal{C} \in (0.066, 0.726)$, where the lowest value corresponds to the MEDLINE graph and the highest to SPIRES (the latter contains publications on high-energy physics). In comparison to these graphs, our mathematical database is more clustered with $\mathcal{C}(G_c) = 0.64$. A more thorough comparison of G_c and the Los Alamos e-Print Archive graph, see Table 5.4. Unfortunately the calculation of the proximity ratio μ is infeasible for graphs of this order, as finding $\mathcal{C}_r$ and $\mathcal{L}_r$ for 30 random graphs of the same order and size is tedious.

Barabási et al. [14] have studied the dynamical properties of collaboration graphs and propose a model to capture the evolution of these graphs. The collaboration graphs they constructed, considering different time periods, contain 70,975 authors for the mathematical data and 209,293 neuroscience authors. They study the evolutionary properties by investigating measures related to authors who have appeared as new vertices in the graphs during some given time period. Using logarithmic binning[6] to reduce noise in the tail they find $\gamma = 2.4$ for the mathematical data and $\gamma = 2.1$ for the neuroscientific data.

An interesting observation is that the average path length $\mathcal{L}$ as well as the clustering coefficient $\mathcal{C}$ of the collaboration networks constructed by Barabási et al. [14] *decreases* in time as the network itself grows. The largest connected component

[6] In order to filter noise in data analysis, data points can be grouped into "bins" either of uniform, linearly growing, or logarithmically growing size.

grows faster than the other parts of the network and the average degree increases. The reasons behind these observations would be of interest to study in future. Their model is essentially a preferential-attachment model of new novice authors combined with preferential introduction of internal edges, which are collaborations between "established" authors. Simplifications are assuming that once a new paper is published (and therefore new edges introduced), there are always a constant number of coauthors, and that the arrival rate of novice authors is constant. We consider implementing and generalizing this model as further work to study the effects of these restrictions on the results reported by Barabási et al. in [14].

5.4 Clustering experiments

We studied the clusters found by a local search using the fitness functions f_1 and f_2 of Equations 4.11 and 4.12 described in Section 4.5.2. We performed simulated annealing (as described in Section 4.5.2) on different graphs $G = (V, E)$, with $\Gamma(v)$ as the initial cluster of vertex v. For a graphical example, see Figure 2.1, where the grouping of the vertices in the picture is done by the clusters found with the fitness function f_1 of Equation 4.11. Edges that connect vertices u and v such that $u \in \mathcal{K}(v)$ and $v \in \mathcal{K}(u)$ are drawn black; edges where only one vertex includes the other in its cluster are drawn dark gray, and plain edges light gray. It would be of interest to know whether this classification of connections has any biological meaning, i.e. whether the black connections are somehow more important than the gray.

We have made the following observations on regular structures for both heuristics. In a K_n or a $K_{n,n}$, each vertex will choose the entire graph as its cluster. For a graph that consists of two cliques of relatively the same size connected by one edge, each vertex chooses its own clique as its cluster. We also clustered a graph that connects a K_{20} and a P_{10} with a single edge, forming a large "head" and a long "tail". All vertices in the "head", including the one to which the tail is connected, consider the clique their cluster. When using f_1, a couple of the tail vertices nearest to the clique also choose parts of the clique to be included in their clusters; after the midpoint of the tail the vertices only consider their immediate neighbors in their cluster. With f_2, none of the tail vertices choose the clique as their cluster, but a couple of their nearest neighbors along the tail.

For a one-level caveman graph, generated as explained in Section 5.1.6 with parameters $n_{\min} = 10$, $n_{\max} = 25$, $p = 0.95$, and $s = 0.95$, both heuristics find exactly the original caves as created by the generation process. The examined instance has $n = 235$ vertices and $m = 2{,}163$ edges and constitutes of 13 caves. We measured how extensively the clustering algorithm traverses the graphs when determining the cluster of a single vertex with simulated annealing using $T_0 = 1{,}000$, $\alpha = 0.95$, taking 25 iterations, each constituting of 100 rounds. The results are shown in Figure 5.14, where the 13 caves are clearly visible due to vertex labeling; the two runs plotted are separate, as the measurement of the explored area consumes some time that does not effect the search behavior; nevertheless the orders of the clusters found remain the same, as the search constitutes of several iterations and hence almost surely finds the local optimum for any run.

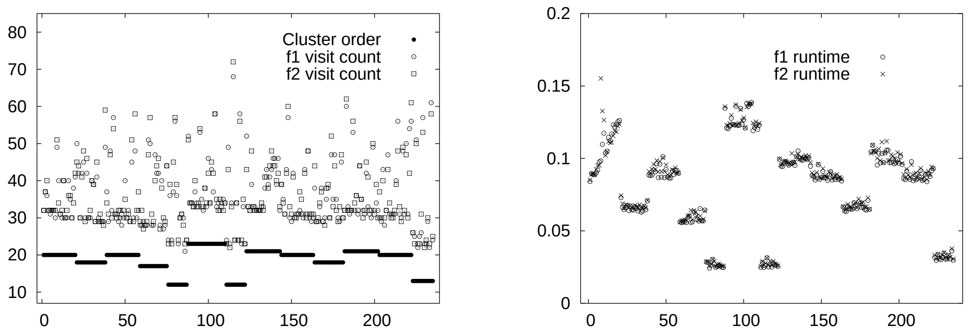
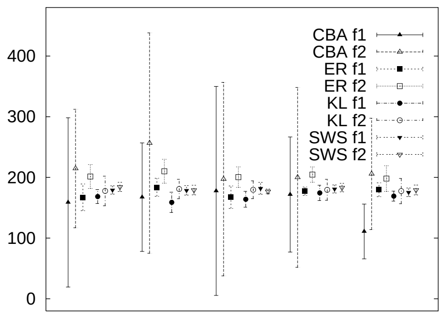


FIGURE 5.14: On the left, the orders of the resulting clusters (same for f_1 and f_2) and number of distinct vertices visited during the search. On the right, the running time of the algorithm per each start vertex for the fitness function f_1 and f_2.

We also chose five instances of each of the test sets of Section 5.2 for all four models CBA ($p = 0.5$), ER, KL ($p = 5$), and SWS ($p = 0.5$), and computed the clusters for 10 randomly chosen vertices in each graph. The graphs have $n = 10{,}000$ and $\delta \approx 0.019$. The average orders of the clusters found and the portions of the graphs traversed during the search are shown in Figure 5.15. The search was run for 30 iterations per start vertex, each iteration having 300 rounds with both fitness functions.

The orders of the clusters found give some hints on the structure of the graphs: in the CBA model, the cluster orders vary significantly, hinting that the vertices are not all equal. The smallest cluster in the CBA sample is the cluster of a vertex of degree 104 and contains only 68 vertices, whereas the largest cluster has order 599 and belongs to a vertex with 628 neighbors. As the minimum degree of all the CBA graphs is 100, the former is clearly not a hub vertex, whereas the latter is of medium degree within the CBA degree distribution, the maximum degree of the CBA instances being above 1,200.

For all of the other models the orders of the clusters vary much less than for the CBA model, suggesting that the vertices in those graphs are more or less equal, which is true by the model descriptions. It seems to be easier to find a cluster in the KL and SWS instances, as the number of vertices examined during the search is only about two thirds of that for the CBA and ER models. This can be explained by the regularity in the structure of the former.



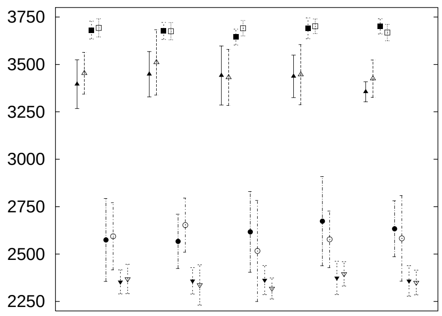


FIGURE 5.15: The averages of the orders of the clusters found (left) and the number of vertices visited during the search (right) starting at 10 random vertices in each graph, 5 graphs from each model CBA, ER, KL, and SWS. The five randomly chosen instances are not ordered in any way and the selection of the random vertices differ for the two fitness functions f_1 and f_2. The standard deviations are shown in both plots; the key is only shown for the left figure and applies for both.

To give some examples of clustering natural data, we have clustered the collaboration graph of Figure 5.12; the outcome of the clustering using f_2 and simulated annealing per each vertex separately is shown below in Figure 5.16.

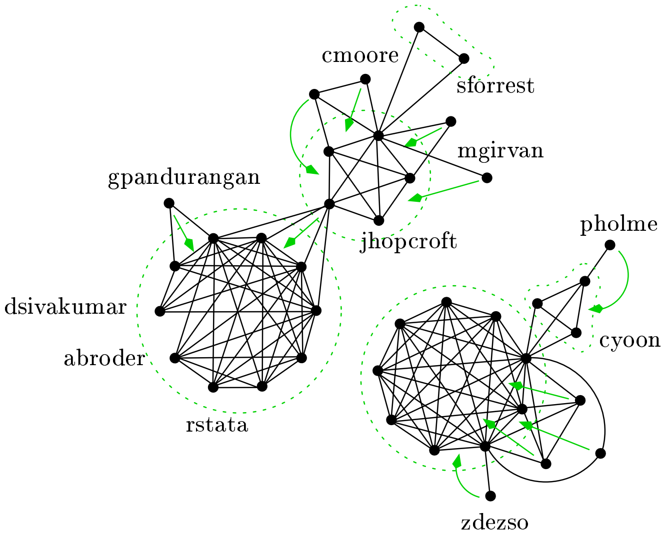


FIGURE 5.16: The clusters of the example collaboration graph; the vertices enclosed in a dotted line all consider each other as members of their cluster, unless an arrow is drawn to indicate the cluster in which the vertex gets grouped by local clustering.

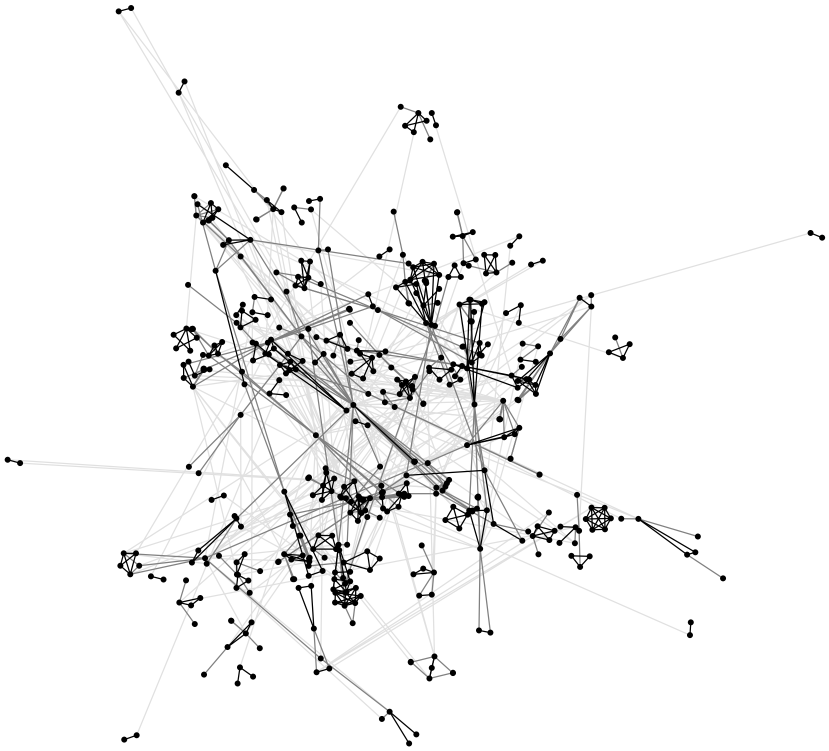


FIGURE 5.17: The clusters of a collaboration graph $G = (V, E)$ with $n = 503$ and $m = 828$, obtained using f_2 and simulated annealing. The edges in E are colored as follows: black edges connect vertices that both consider each other in their cluster, gray those in which only one of the endpoints included the other in its cluster; the rest of the edges in G are drawn light gray. Note that in general $\mathcal{K}(v) \neq \Gamma(v)$ and hence not all of the vertices in $\mathcal{K}(v)$ are connected to v with an edge in E.

We also clustered a larger collaboration graph, shown in Figure 5.17. This figure was produced with a tool implemented by Kosti Rytkönen that uses string forces to hold the graph together with connected vertices as close to each other as possible. Also the *C. elegans* clustering of Figure 2.1 has been drawn with the same tool, but modified by hand. In this larger graph, no manual modification has been made. Even though the visualization tool does not use the information of the clusters in any way, relying only on the adjacency information, it appears to group (in this case and also for many other graphs we have drawn, such as hierarchical caveman graphs of different orders) vertices naturally in a way that brings most clusters (as defined by either f_1 or f_2) physically close.

For comparison, we also provide the unmodified figure of the *C. elegans* neural network clustered using f_2 and drawn with the spring-force method in Figure 5.18. In this case the clusters are joined together to form a “backbone” instead of grouping into small clusters such as those of Figure 5.17.

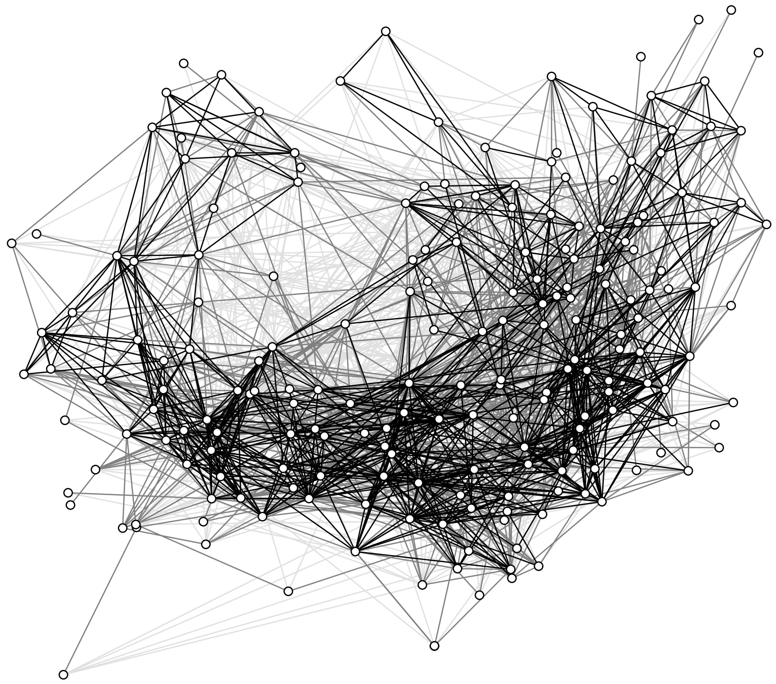


FIGURE 5.18: The *C. elegans* neural network clustered with f_2 and drawn using spring forces. A clustering with f_1 is shown in Figure 2.1 on page 4, with manual modifications on the layout determined by the spring force method.

Scientific collaboration forms a sparse structure in comparison to a nematode brain and hence allows for clearly separate clusters to appear.

We are interested in studying as further work the distributions of cluster orders in larger collaboration graphs, such as those described in Section 5.3, as well as other natural graphs or graph models. Of special interest is clustering the Web graph, for which possible application areas are numerous. Also similarities between the clustering obtained by our local method and clusterings obtained by established global methods are of interest.

Chapter 6

Concluding remarks

The models proposed for generating natural-like networks are numerous, and the simple ideas behind each model can be harnessed further to develop network topology generators that match a particular application area. This development is most evident in modeling efforts of the Internet; communication networks in general and the related algorithms such as routing appear an immediate and fruitful target for design improvements that are founded on observations of network structure.

Our implementations of the models succeed in capturing many of the properties of the models that have been analytically derived, and hence the toolset provides a good foundation for further experimentation and easily extends to cover future models and modifications. Generalizations to weighted and directed graph models are of interest in the future. We especially plan to study further the clustering properties of different graph models as well as natural graphs, aiming to construct formally approachable local clustering algorithms for large graphs. We are also interested in studying methods to obtain random samples from large graphs to avoid the computational difficulty in calculating exact measures for large data sets; studying Markov chains operating on vertex sets of different kinds graphs is of general interest, continuing the study of random walks for different models.

This field of research is still growing. Hence several new proposals for natural-like network models or their essential properties will certainly be published in the future as well. Naturally a new multi-disciplinary research topic such as this will initiate from conjectures and simple studies of limited accuracy, but robust approaches are already starting to appear. There is a strong demand for straightforward analytical approaches, connections to methods of natural sciences, and rigorous experimentation practices. Many promising ideas are currently clouded with incomplete reasoning and experiments of very limited scale. We believe that many useful applications and fruitful discoveries in this area are yet to appear.

Bibliography

[1] E. Aarts and J. K. Lenstra, editors. *Local Search in Combinatorial Optimization*. John Wiley & Sons, New York, NY, USA, 1997.

[2] A.-C. Achilles. The collection of computer science bibliographies. Located at `http://liinwww.ira.uka.de/bibliography/`, accessed December 2, 2002.

[3] L. Adamic. The small world web. In S. Abiteboul and A.-M. Vercoustre, editors, *Proceedings of ECDL'99 in Paris, France*, volume 1696 of *Lecture Notes in Computer Science*, pages 443–452. Springer-Verlag, Berlin, Germany, 1999.

[4] L. A. Adamic and B. A. Huberman. Power-law distribution of the World Wide Web. *Science*, 287:2115, 2000. Techical comment on [12] with a response from Barabási et al.

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

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

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

[8] L. A. N. Amaral, A. Scala, M. Barthelemy, and H. E. Stanley. Classes of small-world networks. *Proceedings of the National Academy of Sciences, USA*, 97(21):11149–11152, 2000.

[9] M. Arita. Graph modeling of metabolism. *Journal of Japanese Society for Artificial Intelligence*, 15(4):703–710, 2000.

[10] F. G. Ball, D. Mollison, and G. Scalia-Tomba. Epidemics with two levels of mixing. *Annals of Applied Probability*, 7(1):46–89, 1997.

[11] A.-L. Barabási. *Linked: The New Science of Networks*. Perseus Publishing, Cambridge, MA, USA, 2002.

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

[13] A.-L. Barabási, R. Albert, and H. Jeong. Mean-field theory for scale-free random networks. *Physica A*, 272:173–187, 1999.

[14] A.-L. Barabási, H. Jeong, Z. Néda, E. Ravasz, A. Schubert, and T. Vicsek. Evolution of the social network of scientific collaborations. *Physica A*, 311(3–4):590–614, 2002.

[15] A.-L. Barabási, E. Ravasz, and T. Vicsek. Deterministic scale-free networks. *Physica A*, 299(3–4):559–564, 2001.

[16] A. D. Barbour and G. Reinert. Small worlds. *Random Structures and Algorithms*, 19(1):54–74, 2001.

[17] M. Barth and L. A. N. Amaral. Small-world networks: Evidence for a crossover picture. *Physical Review Letters*, 82(15):3180–3183, 1999.

[18] E. Behrends. *Introduction to Markov Chains, with Special Emphasis on Rapid Mixing*. Vieweg & Sohn, Braunschweig, Germany, 2000.

[19] B. Bollobás. *Random Graphs*, volume 73 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, UK, second edition, 2001.

[20] B. Bollobás, O. Riordan, J. Spencer, and G. Tusnády. The degree sequence of a scale-free random graph process. *Random Structures and Algorithms*, 18(3):279–290, 2001.

[21] B. Bollobás and O. M. Riordan. The diameter of a scale-free random graph. Submitted for publication in *Combinatorica*, available online at `http://www.dpmms.cam.ac.uk/~omr10/`.

[22] I. M. Bomze, M. Budinich, P. M. Pardalos, and M. Pelillo. The maximum clique problem. In D.-Z. Du and P. M. Pardalos, editors, *Handbook of Combinatorial Optimization*, volume Suppl. A, pages 1–74. Kluwer Academic Publishers, Boston, MA, USA, 1999.

[23] A. Broder, S. R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the web. *Computer Networks*, 33(1–6):309–320, 2000.

[24] T. Bu and D. Towsley. On distinguishing between internet power law topology generators. In *IEEE Infocom: The 21st Annual Joint Conference of the IEEE Computer and Communications Societies in New York, NY, USA*. IEEE Computer Society Press, Los Alamitos, CA, USA, 2002.

[25] K. L. Calvert, M. B. Doar, and E. W. Zegura. Modeling Internet topology. *IEEE Communications Magazine*, 35(6):160–163, 1997.

[26] F. Chung and L. Lu. The average distances in random graphs with given expected degrees. *Proceedings of the National Academy of Sciences, USA*, 99(25):15879–15882, 2002.

[27] F. R. K. Chung. *Spectral Graph Theory.* American Mathematical Society, Providence, RI, USA, 1997.

[28] R. Cohen, K. Erez, D. ben Avraham, and S. Havlin. Resilience of the Internet to random breakdowns. *Physical Review Letters*, 85(21):4626–4628, 2000.

[29] F. Comellas, J. Ozón, and J. G. Peters. Deterministic small-world communication networks. *Information Processing Letters*, 76(1–2):83–90, 2000.

[30] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms.* McGraw-Hill Book Co., Boston, MA, USA, second edition, 2001.

[31] M. S. Corson and A. Ephremides. A distributed routing algorithm for mobile wireless networks. *Wireless Networks*, 1(1):61–81, 1995.

[32] M. E. Crovella, M. Harchol-Balter, and C. D. Murta. Task assignment in a distributed system: Improving performance by unbalancing load. Technical Report BUCS-TR-1997-018, Department of Computer Science, Boston University, 1997.

[33] J. Davidsen, H. Ebel, and S. Bornholdt. Emergence of a small world from local interactions: Modeling acquaintance networks. *Physical Review Letters*, 88(12):128701, 2002.

[34] N. Deo and P. Gupta. Sampling the web graph with random walks. *Congressus Numerantium*, 149:143–154, 2001.

[35] Z. Dezső and A.-L. Barabási. Halting viruses in scale-free networks. *Physical Review E*, 65(5), 2002.

[36] D. Dhyani, W. K. Ng, and S. S. Bhowmick. A survey of web metrics. *ACM Computing Surveys*, 34(4):469–503, 2002.

[37] R. Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics.* Springer-Verlag, New York, NY, USA, 2000. Full text available online at `http://www.math.uni-hamburg.de/home/diestel/books/graph.theory/`.

[38] M. Doar and I. M. Leslie. How bad is naive multicast routing? In *IEEE Infocom: The Twelfth Annual Joint Conference of the IEEE Computer and Communications Societies in San Francisco, CA, USA*, volume 1, pages 82–89. IEEE Computer Society Press, Los Alamitos, CA, USA, 1993.

[39] M. B. Doar. A better model for generating test networks. In *GLOBECOM '96: IEEE Global Telecommunications Conference in London, UK*. IEEE, Piscataway, NJ, USA, 1996.

[40] S. N. Dorogovtsev, A. V. Goltsev, and J. F. F. Mendes. Pseudofractal scale-free web. *Physical Review E*, 65(6):066122, 2002.

[41] S. N. Dorogovtsev and J. F. F. Mendes. Evolution of networks. *Advances in Physics*, 51(4):1079–1187, 2002.

[42] S. N. Dorogovtsev, J. F. F. Mendes, and A. N. Samukhin. Structure of growing networks with preferential linking. *Physical Review Letters*, 85(21):4633–4636, 2000.

[43] P. Erdős and A. Rényi. On random graphs I. In *Selected papers of Alfréd Rényi*, volume 2, pages 308–315. Akadémiai Kiadó, Budapest, Hungary, 1976. First publication in 1959.

[44] P. Erdős and A. Rényi. On the evolution of random graphs. In *Selected papers of Alfréd Rényi*, volume 2, pages 482–525. Akadémiai Kiadó, Budapest, Hungary, 1976. First publication in 1960.

[45] K. A. Eriksen and M. Hörnquist. Scale-free growing networks imply linear preferential attachment. *Physical Review E*, 65(1):017102, 2002.

[46] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the Internet topology. In *Proceedings of the ACM SIGCOMM '99 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication in Cambridge, MA, USA*, pages 251–262. ACM Press, New York, NY, USA, 1999.

[47] I. J. Farkas, I. Derényi, H. Jeong, Z. Néda, Z. N. Oltvai, E. Ravasz, A. Schubert, A.-L. Barabási, and T. Vicsek. Networks in life: scaling properties and eigenvalue spectra. *Physica A*, 314(1–4):25–34, 2002.

[48] P. Flajolet, S. N. Kostas Hatzis, and P. Spirakis. On the robustness of interconnections in random graphs: a symbolic approach. *Theoretical Computer Science*, 287(2):515–534, 2002.

[49] R. W. Floyd. Algorithm 97: Shortest path. *Communications of the ACM*, 5(6):345, 1962.

[50] A. M. Frieze and C. McDiarmid. Algorithmic theory of random graphs. *Random Structures and Algorithms*, 10(1–2):5–42, 1997.

[51] A. M. Frieze and M. Molloy. Broadcasting in random graphs. *Discrete Applied Mathematics*, 54:77–79, 1994.

[52] M. R. Garey and D. S. Johnson. *Computers and Intractability A Guide to the Theory of NP-Completeness.* W. H. Freeman, San Francisco, CA, USA, 1979.

[53] I. P. Gent, H. H. Hoos, P. Prosser, and T. Walsh. Morphing: Combining structure and randomness. In *AAAI/IAAI 99: Proceedings of the 16th National Conference on Artificial Intelligence and 11th Conference on on Innovative Applications of Artificial Intelligence in Orlando, Florida, USA*, pages 654–660. AAAI Press / The MIT Press, Menlo Park, CA, USA, 1999.

[54] E. N. Gilbert. Random graphs. *Annals of Mathematical Statistics*, 30(4):1141–1144, 1959.

[55] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences, USA*, 99(12):7821–7826, 2002.

[56] P. M. Gleiss, P. F. Stadler, A. Wagner, and D. A. Fell. Small cycles in small worlds. Working paper 0-10-058, Santa Fe Institute, 2000.

[57] K.-I. Goh, B. Kahng, and D. Kim. Spectra and eigenvectors of scale-free networks. *Physical Review E*, 64(5):051903, 2001.

[58] C. P. Gomes and B. Selman. Problem structure in the presence of perturbations. In *AAAI/IAAI 97: Proceedings of the 14th National Conference on Artificial Intelligence and Ninth Innovative Applications of Artificial Intelligence Conference in Providence, RI, USA*, pages 221–226. AAAI Press / The MIT Press, Menlo Park, CA, USA, 1997.

[59] C. P. Gomes, B. Selman, and H. A. Kautz. Boosting combinatorial search through randomization. In *AAAI/IAAI 98: Proceedings of the 15th National Conference on Artificial Intelligence and Tenth Innovative Applications of Artificial Intelligence Conference in Madison, Wisconsin, USA*, pages 431–437. AAAI Press / The MIT Press, Menlo Park, CA, USA, 1998.

[60] G. R. Grimmett. *Percolation.* Springer-Verlag, Berlin, Germany, second edition, 1999.

[61] G. R. Grimmett. Percolation. In J.-P. Pier, editor, *Development of Mathematics 1950–2000*, pages 547–576. Birkhäuser, Boston, MA, USA, 2000.

[62] G. R. Grimmett and D. R. Stirzaker. *Probability and Random Processes.* Oxford University Press, Oxford, UK, third edition, 2001.

[63] J. Guare. *Six Degrees of Separation: A Play.* Vintage, New York, NY, USA, 1990.

[64] E. Hartuv and R. Shamir. A clustering algorithm based on graph connectivity. *Information Processing Letters*, 76(4–6):175–181, 2000.

[65] P. Holme and B. J. Kim. Growing scale-free networks with tunable clustering. *Physical Review E*, 65(2):026107, 2002.

[66] B. A. Huberman and L. A. Adamic. Growth dynamics of the world-wide web. *Nature*, 401:131–132, 1999.

[67] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264 – 323, 1999.

[68] H. Jeong, Z. Néda, and A.-L. Barabási. Measuring preferential attachment for evolving networks. *Europhysics Letters*, 61(4):567–572, 2003.

[69] 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.

[70] C. Jin, Q. Chen, and S. Jamin. Inet: Internet topology generator. Technical Report CSE-TR443-00, Department of EECS, University of Michigan, 2000.

[71] S. Jung, S. Kim, and B. Kahng. A geometric fractal growth model for scale-free networks. *Physical Review E*, 65(6):056101, 2002.

[72] D. Jungnickel. *Graphs, Networks and Algorithms.* Springer-Verlag, Berlin, Germany, 1999.

[73] S. Kapidakis. *Average-Case Analysis of Graph-Searching Algorithms.* PhD thesis, Department of Computer Science, Princeton University, 1990.

[74] R. Kasturirangan. Multiple scales in small-world networks. Technical Report AIM-1663, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, 1999.

[75] B. J. Kim, C. N. Yoon, S. K. Han, and H. Jeong. Path finding strategies in scale-free networks. *Physical Review E*, 65(2):027103, 2002.

[76] S. Kim. Graph theoretic sequence clustering algorithms and their applications to genome comparison. In J. T. L. Wang, C. H. Wu, and P. P. Wang, editors, *Computational Biology and Genome Informatics*, chapter 4. World Scientific Publishing Company, Singapore, 2003. To appear.

[77] S. Kirkpatrick, C. D. G. Jr. and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

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

[79] J. M. Kleinberg. The small-world phenomenon: an algorithm perspective. In *STOC: Proceedings of the 32nd Annual ACM Symposium on Theory of Computing in Portland, OR, USA*, pages 163–170. ACM Press, New York, NY, USA, 2000.

[80] J. M. Kleinberg, S. R. Kumar, P. Raghavan, S. Rajagopalan, and A. S. Tomkins. The Web as a graph: Measurements, models, and methods. In T. Asano, H. Imai, D. Lee, S. Nakano, and T. Tokuyama, editors, *Proceedings of the Fifth Annual International Conference on Computing and Combinatorics in Tokyo, Japan*, volume 1627 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, Germany, 1999.

[81] J. Kleinfeld. Six degrees of separation: Urban myth? *Psychology Today*, 2002.

[82] D. E. Knuth. *Seminumerical Algorithms*, volume 2 of *The Art of Computer Programming*. Addison-Wesley, Reading, Massachusetts, third edition, 1997.

[83] P. L. Krapivsky, S. Redner, and F. A. Leyvraz. Connectivity of growing random networks. *Physical Review Letters*, 85(21):4629–4632, 2000.

[84] S. R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal. Stochastic models for the Web graph. In *FOCS: 41st Annual Symposium on Foundations of Computer Science in Redondo Beach, CA, USA*, pages 57–65. IEEE Computer Society Press, Los Alamitos, CA, USA, 2000.

[85] S. R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Extracting large-scale knowledge bases from the web. In *Proceedings of International Conference on Very Large Data Bases in Edinburgh, Scotland, UK*, pages 639–650. Morgan Kaufmann Publishers, San Francisco, CA, USA, 1999.

[86] V. Latora and M. Marchiori. Efficient behavior of small-world networks. *Physical Review Letters*, 87(19):198701, 2001.

[87] S. Lawrence and C. L. Giles. Accessibility of information on the web. *Nature*, 400:117–119, 1999.

[88] M. Levene, T. Fenner, G. Loizou, and R. Wheeldon. A stochastic model for the evolution of the web. *Computer Networks*, 39:277–287, 2002.

[89] T. Łuczak. Phase transition phenomena in random discrete structures. *Discrete Mathematics*, 136(1–3):225–242, 1994.

[90] M. Marchiori and V. Latora. Harmony in the small-world. *Physica A*, 285(3–4):539–546, 2000.

[91] H. Matsuda, T. Ishihara, and A. Hashimoto. Classifying molecular sequences using a linkage graph with their pairwise similarities. *Theoretical Computer Science*, 210(2):305–325, 1999.

[92] A. Medina, I. Matta, and J. Byers. On the origin of power laws in Internet topologies. *ACM Computer Communication Review*, 30(2):18–28, 2000.

[93] A. Mehrotra and M. A. Trick. A column generation approach for graph coloring. *INFORMS Journal on Computing*, 8:344–354, 1996.

[94] M. Mihail, C. Gkantsidis, A. Saberi, and E. Zegura. On the semantics of Internet topologies. Technical Report GIT-CC-02-07, College of Computing, Georgia Institute of Technology, 2002.

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

[96] M. Molloy and B. Reed. A critical point for random graphs with a given degree sequence. *Random Structures and Algorithms*, 6:161–180, 1995.

[97] C. Moore and M. E. J. Newman. Epidemics and percolation in small-world networks. *Physical Review E*, 61(5):5678–5682, 2000.

[98] A. E. Motter, A. P. S. de Moura, Y.-C. Lai, and P. Dasgupta. Topology of the conceptual network of language. *Physical Review E*, 65(6):065102, 2002.

[99] G. Mukherjee and S. S. Manna. Quasistatic scale-free networks. *Physical Review E*, 67(1):012101, 2003.

[100] M. E. J. Newman. Clustering and preferential attachment in growing networks. *Physical Review E*, 64(2):025102, 2001.

[101] M. E. J. Newman. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences, USA*, 98(2):404–409, 2001.

[102] M. E. J. Newman, S. Forrest, and J. Balthrop. Email networks and the spread of computer viruses. *Physical Review E*, 66(3):035101, 2002.

[103] M. E. J. Newman and M. Girvan. Mixing patterns and community structure in networks. In R. Pastor-Satorras and J. Rubi, editors, *Proceedings of the XVIII Sitges Conference on Statistical Mechanics in Barcelona, Spain*, Lecture Notes in Physics. Springer-Verlag, Berlin, Germany, 2002. To appear in 2003.

[104] M. E. J. Newman, C. Moore, and D. J. Watts. Mean-field solution of the small-world network model. *Physical Review Letters*, 84(14):3201–3204, 2000.

[105] M. E. J. Newman, S. H. Strogatz, and D. J. Watts. Random graphs with arbitrary degree distributions and their applications. *Physical Review E*, 64(2):026118, 2001.

[106] M. E. J. Newman and D. J. Watts. Scaling and percolation in the small-world network model. *Physical Review E*, 60(6):7332–7342, 1999.

[107] M. E. J. Newman, D. J. Watts, and S. H. Strogatz. Random graph models of social networks. *Proceedings of the National Academy of Sciences, USA*, 99(Suppl. 1):2566–2572, 2002.

[108] S. Niskanen and P. R. J. Östergård. Cliquer user's guide, version 1.0. Technical Report T48, Communications Laboratory, Helsinki University of Technology, 2003.

[109] E. Nuutila. Efficient transitive closure computation in large digraphs. *Acta Polytechnica Scandinavica: Mathematics and Computing in Engineering*, 74:124, 1995. Doctoral thesis, Helsinki University of Technology, Department of Computer Science.

[110] J. Ozón Górriz. *Contribución al coloreado de grafos y las redes pequeño-mundo*. PhD thesis, Universitat Politècnica de Catalunya, 2001.

[111] S. A. Pandit and R. E. Amritkar. Characterization and control of small-world networks. *Physical Review E*, 60(2):R1119–R1122, 1999.

[112] G. Pandurangan, P. Raghavan, and E. Upfal. Building low-diameter P2P networks. In *FOCS: 42nd Annual Symposium on Foundations of Computer Science in Las Vegas, Nevada*, pages 492–499. IEEE Computer Society Press, Los Alamitos, CA, USA, 2001.

[113] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, New York, NY, USA, 1994.

[114] R. Pastor-Satorras and A. Vespignani. Epidemic spreading in scale-free networks. *Physical Review Letters*, 86(14):3200–3203, 2001.

[115] V. Paxson and S. Floyd. Why we don't know how to simulate the internet. In *Proceedings of the 1997 Winter Simulation Conference in Atlanta, Georgia, USA*, pages 1037–1044. ACM Press, New York, NY, USA, 1997.

[116] P. Raghavan. Discussion at the Isaac Newton Institute for Mathematical Sciences, Cambridge, UK, 2002. Workshop on Topics in Computer Communication and Networks.

[117] E. Ravasz and A.-L. Barabási. Hierarchical organization in complex networks. *Physical Review E*, 67(2):026112, 2003.

[118] S. Redner. How popular is your paper? an empirical study of the citation distribution. *European Physical Journal B*, 4(2):131–134, 1998.

[119] R. Sedgewick and P. Flajolet. *An Introduction to the Analysis of Algorithms*. Addison-Wesley, Reading, MA, USA, 1996.

[120] B. Shargel, H. Sayama, I. R. Epstein, and Y. Bar-Yam. Optimization of robustness and connectivity in complex networks. *Physical Review Letters*, 90(6):068701, 2003.

[121] H. A. Simon. On a class of skew distribution functions. *Biometrika*, 42(3/4):425–440, 1955.

[122] P. R. J. Östergård. A fast algorithm for the maximum clique problem. *Discrete Applied Mathematics*, 120(1–3):197–207, 2002.

[123] B. Tadić. Adaptive random walks on the class of web graphs. *European Physics Journal B*, 23(2):221–228, 2001.

[124] B. Tadić. Dynamics of directed graphs: the world-wide web. *Physica A*, 293(1–2):273–284, 2001.

[125] B. Tadić. Growth and structure of the world-wide web: Towards realistic modeling. *Computer Physics Communications*, 147(1–2):586–589, 2002.

[126] B. Tadić. Temporal fractal structures: origin of power laws in the world-wide web. *Physica A*, 314(1–4):278–283, 2002.

[127] D. Volchenkov and P. Blanchard. An algorithm generating random graphs with power law degree distributions. *Physica A*, 315(3–4):677–690, 2002.

[128] D. Vukadinović, P. Huang, and T. Erlebach. On the spectrum and structure of internet topology graphs. In H. Unger, T. Böhme, and A. Mikler, editors, *Proceedings of Second International Workshop on Innovative Internet Computing Systems (IICS 2002) in Kühlungsborn, Germany*, volume 2346 of *Lecture Notes in Computer Science*, pages 83–95. Springer-Verlag, Berlin, Germany, 2002.

[129] A. Vázquez, R. Pastor-Satorras, and A. Vespignani. Large-scale topological and dynamical properties of the internet. *Physical Review E*, 65(6):066130, 2002.

[130] A. Vázquez and M. Weigt. Computational complexity arising from degree correlations in networks. *Physical Review E*, 67(2):027101, 2003.

[131] T. Walsh. Search in a small world. In *IJCAI'99: Proceedings of the 16th International Joint Conference on Artificial Intelligence in Stockholm, Sweden*, volume 2, pages 1172–1177. Morgan Kaufmann Publishers, San Francisco, CA, USA, 1999.

[132] T. Walsh. Search on high degree graphs. In *IJCAI'01: Proceedings of the 17th International Joint Conference on Artificial Intelligence in Seattle, Washington, USA*, pages 266–274. Morgan Kaufmann Publishers, San Francisco, CA, USA, 2001.

[133] D. J. Watts. *Small Worlds*. Princeton University Press, Princeton, NJ, USA, 1999.

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

[135] B. M. Waxman. Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications*, 6(9):1617–1622, 1988.

[136] M. Weigt. Dynamics of heuristic optimization algorithms on random graphs. *The European Physical Journal B*, 28(3):369–381, 2002.

[137] E. W. Zegura, K. L. Calvert, and S. Bhattacharjee. How to model an internetwork. In *IEEE Infocom: The 15th Annual Joint Conference of the IEEE Computer and Communications Societies in San Francisco, CA, USA*, volume 2, pages 594–602. IEEE Computer Society Press, Los Alamitos, CA, USA, 1996.

[138] E. W. Zegura, K. L. Calvert, and M. J. Donahoo. A quantitative comparison of graph-based models for Internet topology. *IEEE / ACM Transactions on Networking*, 5(6):770–783, 1997.

[139] H. Zhang, A. Goel, and R. Govindan. Using the small world model to improve Freenet performance. In *IEEE Infocom: The 21st Annual Joint Conference of the IEEE Computer and Communications Societies in New York, NY, USA*. IEEE Computer Society Press, Los Alamitos, CA, USA, 2002.

[140] G. K. Zipf. *Human behavior and the principle of least effort*. Addison-Wesley Press, Cambridge, MA, USA, 1949.