World Wide Web:
A Graph-Theoretic Perspective

Narsingh Deo
deo@cs.ucf.edu

Pankaj Gupta
pgupta@cs.ucf.edu

School of Computer Science
University of Central Florida
Orlando, FL 32816
USA

University of Central Florida
Computer Science Technical report
CS-TR-01-001
March 2001

ABSTRACT

The World Wide Web can be modeled as a directed graph in which a node represents a Web page and an edge represents a hyperlink. Currently, the number of nodes in this gigantic Web graph is estimated to be over four billion, and is growing at more than seven million nodes a day — without any centralized control. Recent studies suggest that despite its chaotic appearance, the Web is a highly structured digraph, in a statistical sense. The study of this graph can provide insight into Web algorithms for crawling, searching, and ranking Web resources. Knowledge of the graph-theoretic structure of the Web graph can be exploited for attaining efficiency and comprehensiveness in Web navigation as well as enhancing Web tools, *e.g.*, better search engines and intelligent agents. In this proposal, we discuss various problems to be explored for understanding the structure of the WWW. Many research directions are identified such as Web caching, prevention of security threats, user-flow analysis, *etc*.

TABLE OF CONTENTS

1. INTRODUCTION 1

2. MODELING THE WWW 3

2.1 WWW IS NOT THE INTERNET 3
2.2 WWW AS A DIGRAPH 5

3. RANDOM-GRAPH MODELS 7

3.1 ERDÖS-RÉNYI MODEL 7
3.2 SMALL-WORLD MODELS 9
3.3 THE PREFERENTIAL-ATTACHMENT MODEL 15
3.4 THE WEB-SITE GROWTH MODEL 18
3.5 AN EVOLVING WEB-GRAPH MODEL 21
3.6 THE ALPHA-BETA MODEL 26

4. EMPIRICAL STUDIES 26

4.1 THE BOWTIE EMPIRICAL STUDY 27
4.2 SAMPLING THE WEB THROUGH RANDOM WALK 30

5. PROPERTIES OF THE WEB GRAPH 33

5.1 POWER-LAW DISTRIBUTION 34
5.2 DIAMETER OF THE WEB GRAPH 35
5.3 CONNECTED COMPONENTS 35
5.4 SIZE AND GROWTH CHARACTERISTICS 36
5.5 CHANGING CONTENTS OF A WEB PAGE 37

6. SEARCH ENGINES 38

6.1 ARCHITECTURE OF A SEARCH ENGINE 39
6.2 ISSUES IN THE DESIGN OF EFFECTIVE SEARCH-ENGINES 41

7. GRAPH-THEORETIC WEB ALGORITHMS 42

7.1 THE WWW COMMUNITIES 42
7.2 HITS ALGORITHM 44
7.3 MINING KNOWLEDGE-BASES 47
7.4 WEB-PAGE EVALUATION 49
7.4.1 PageRank Algorithm 49
7.4.2 Page Reputation 51
7.4.3 Markov-Chain-Based Rank Method 53
7.5 SMALL-WORLD ALGORITHMICS 56
7.6 RELATED-URL AND TOPIC DISTILLATION ALGORITHMS 57

8. CONCLUSION 61

REFERENCES 63

APPENDIX I — DEFINITIONS, ABBREVIATIONS, AND SYMBOLS 69

LIST OF SYMBOLS 71

1. INTRODUCTION

The World Wide Web (WWW or Web) has revolutionized the way we access information. By April 2001, the Web is estimated to have over 4 billion pages, more than 28 billion hyperlinks, and is growing rapidly at the rate of 7.3 million pages a day (Moore and Murray [49], Kleinberg *et al.* [39]). This gigantic structure of the Web often makes it difficult for even the most technical users to find the best information available on a given topic. The WWW can be viewed in two very divergent ways. From an internal standpoint, it is a TCP-compliant application, HTTP with related software, *e.g.*, support for Java and a set of associated programming, and data communications problems and solutions. We are interested in the other equally important way of viewing the WWW, which is external and extensional. From this standpoint, the Web is a vast and continuously growing repository of information: textual as well as audio and video. The size, number of pages, and presence of hyperlinks distinguish the Web from a distributed database. The sheer volume of material and the apparently chaotic nature of the Web can make locating, acquiring, and organizing information for a specific purpose both time consuming and difficult. This runs exactly opposite to the promise of the Web, namely that all information can be made readily available to the world's population all of the time. The first challenge to overcome in attaining this laudable goal would be to find methods for efficiently and comprehensively navigating the WWW. Efficiency has to be defined in WWW-intrinsic terms. That is, one wants to minimize the number of links that must be followed to reach a desired piece of information from a designated starting point. Com-

prehensiveness means that most relevant information is actually obtained during the navigation. At present, neither efficiency nor comprehensiveness can be reliably achieved.

Much of the effort in devising search methods for navigating the WWW would be based on knowledge engineering. This is a promising approach provided one knows how deep (how many links overall) the search should continue and how comprehensively relevant sections of the WWW have been identified. Knowledge engineering is most appropriate when the field of search has been significantly narrowed to the point where with some certainty most of the relevant information resides in visited pages. At this point, knowledge-engineering techniques can be used to amplify relevance and filter less relevant material away so that the user can then apply interface techniques, *e.g.*, visualization to examine the results. There is need of a set of methods that can exploit structural properties of the WWW to accelerate the search process, so that it actually reaches the threshold where knowledge engineering can be profitably applied. A summary of the WWW data collection and characterizations of aspects of the WWW appears in Pitkow [56]. However, a Web model does not 'emerge' from these data, as the elements of any model must be formulated mathematically.

Despite its chaotic appearance, the Web is highly structured, but in a statistical sense. Models have been proposed which reproduce certain experimentally determined features of the WWW and these features could be exploited to attain efficiency and comprehensiveness in the WWW navigation. Graph theory aids in understanding the structure of the Web at macroscopic as well as microscopic level. An overview of applications of graph theory to the WWW appears in Hayes [33, 34]. In this paper, we survey the present re-

search on Web modeling, search algorithms, and properties of the Web from a graph-theoretic perspective. The paper is organized as follows. Section 2 discusses some preliminary considerations about the WWW that play a crucial role in assessing its models. Section 3 is a survey of the random-graph models that explain the Web structure. Section 4 covers various empirical studies of the WWW. Section 5 discusses properties of the Web that have been discovered until now. Section 6 points out features of the present search engines and analyzes factors for improving them. Section 7 describes algorithms applied for search and identification of Web communities. Section 8 discusses the proposed work. Section 9 presents the concluding remarks. Appendix I provides the definitions of the important graph-theoretic terms as well as abbreviations used. The description of the graph-theoretic terms used in this paper can be found in a standard graph-theory book such as [24].

2. MODELING THE WWW

An accurate model of the Web is helpful for testing the correctness and scalability of a Web algorithm. In addition, modeling the Web structure will enable us to predict its future properties likely to emerge from its current pattern of evolution.

2.1 WWW is not the Internet

Any discussion of the WWW in the extensional sense must begin by disengaging it from the Internet. The Internet is a constellation of data communications technologies, which are not relevant for the Web structure. There is no correspondence between the distance

(minimum number of hyperlinks) from one Web page to another and the minimum number of IP router hops required to realize the traversal. It may be a three-hyperlink jump for navigating from one Web page to another, but nine routers might have to be involved in the best of circumstances on the Internet. Internet topology and traffic are defined by data communications lines and resource allocation issues in entities like routers or operating systems. Claffy [21] and Crovella *et al.* [22] presented valuable studies of the Internet and the impact certain kinds of WWW activities have on it as a resource issue. However, the WWW and the Internet are two very distinct entities.

The smallest useful component of the WWW is a Web page. A page can be resolved into smaller elements such as cross-linked subdocuments (HTML files) and multimedia objects, *e.g.*, digital images. This level of detail is not necessary for an analysis of the Web structure. The design of an individual Web page and its outgoing and incoming hyperlinks are not influenced by data communications or systems considerations. Since, the WWW is the sum total of the authoring decisions made by individual designers of a page, the hyperlinks on a Web page reflect semantically motivated, intentional acts by human beings. Recent studies about the WWW structure show that it resembles collective behavior reminiscent of complex physical systems and thermodynamics. Thus, despite the impossibility of direct modeling at the microscopic level, the WWW structure can be modeled. Such model will be used to improve existing search methods and invent new scalable approaches that can match the growth of the Web.

The Web resembles a distributed database superficially. However, it has many features that distinguish it from a distributed database. The uncontrolled, decentralized, and

rapid growth of the Web is absent in a distributed database. Web pages are constantly being created, maintained, and modified by hundreds of thousands, perhaps millions, of users all over the world. The Web is unique in the sense that there is no uniform structure, integrity constraints, transactions, standard query language, or a data model [29]. Moreover, the key features of a database such as reliability, recovery, *etc*., are not present in the Web.

2.2 WWW as a Digraph

The WWW can be modeled as a directed graph where each node represents a page and each edge, a hyperlink. We refer the directed graph formed by the WWW as the *Web graph*. A small section of the Web graph is shown in Figure 1.

To draw one more distinction between the Internet and the WWW, the word ‘site’ has various meanings when applied to the WWW. From the Internet standpoint, a site is a principal IP address. It can be identified as a single, reachable target on the Internet. From the WWW standpoint, a Web site is defined as a registered domain-name on the Internet. As an interesting exercise in both WWW search and the framing of the definition, here are some URLs, that have conflicting definitions of the term Web site (FOLDOC [62], Netlingo [63], Webopedia [64]). Whatever appropriate definition eventuates, it does not coincide with IP site.

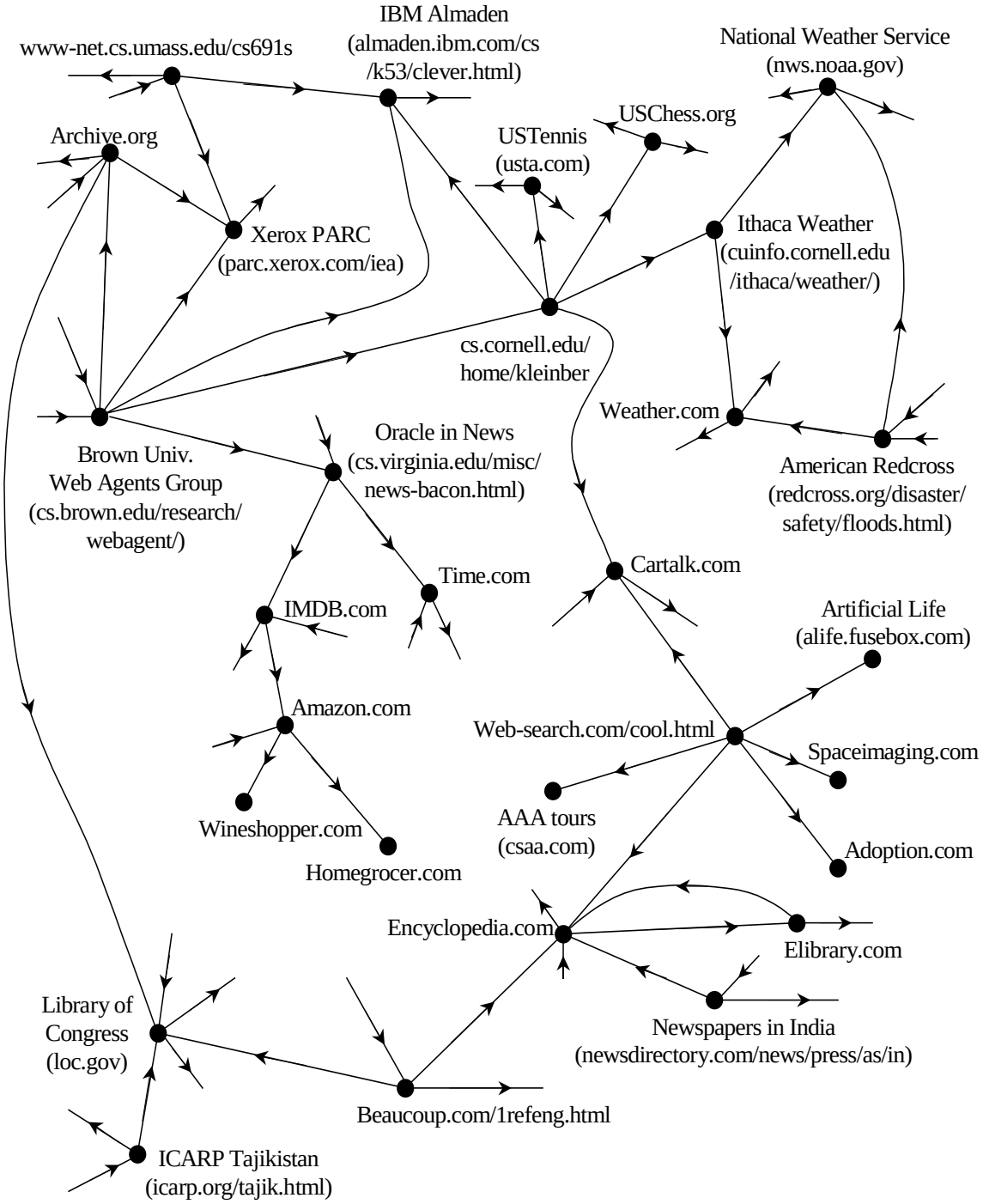


Figure 1. A Small Section of the Web Graph

3. RANDOM-GRAPH MODELS

This section surveys various models of random graphs that have been proposed for the WWW. We also point out drawbacks of these models. Some of these models capture the growth characteristics of the Web, while others are only static.

3.1 Erdös-Rényi Model

Erdös and Rényi [28] proposed the earliest model of a random graph. The model starts with a null graph having n nodes. Each of the $\frac{n \cdot (n-1)}{2}$ pairs of nodes is connected with an edge at random with a specified uniform probability p. At a threshold value p_c of the probability p, many interesting properties appear in the graph. When $p < p_c$, ($p_c \sim 1/n$), the graph has many disconnected components. At $p = p_c$, a large connected component is formed, which in the asymptotic limit contains all the nodes in the graph. The number of nodes in this model is fixed. Therefore, this model cannot capture the growth characteristics of the Web. In addition, a uniform probability p of edge formation between all pairs of nodes does not truly reflect the real-life WWW.

An example of the Erdös-Rényi model is shown in Figure 2(a) and 2(b). Initially, we have a null graph with $n = 10$ nodes as in Figure 2(a). With a probability $p = 0.2$, an edge is added between any two nodes, *e.g.*, nodes c and j. This process is repeated until a total of $m = \frac{p \cdot n \cdot (n-1)}{2} = 9$ edges are added, as shown in Figure 2(b). This leads to the formation of a random graph, which has a small diameter.

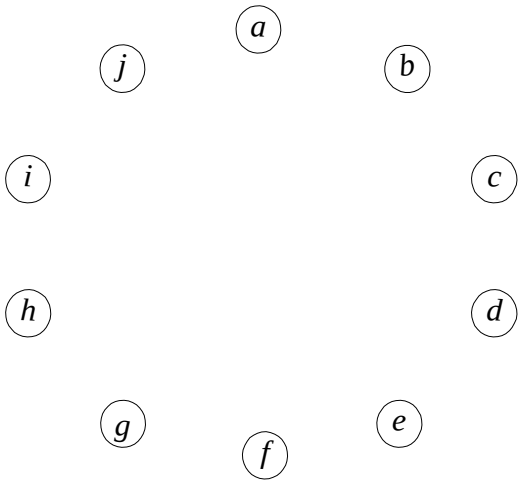


Figure 2(a). A Null Graph (n = 10)

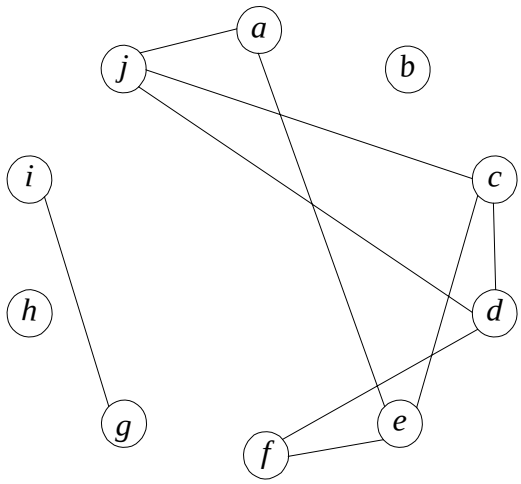


Figure 2(b). Erdös-Rényi Random Graph With 10 Nodes and 9 Edges

3.2 Small-World Models

The small-world model of random networks has inspired much of the research into the WWW structure. Milgram [48] appears to be the first to articulate the basic properties of small-world networks. It is a well-founded folklore that each individual is indirectly linked through a short chain of acquaintances to practically anyone else in the world. Sparseness, small diameter, and cliquishness[1] are the three properties that characterize small-world networks. Erdös-Rényi type random graphs have small diameter, but they lack the cliquishness present in small-world networks. The small-world effect has been identified in disparate contexts including neural network of the worm *C. elegans*, epidemiology [59], power-grid network, collaboration graph of film actors (*e.g.*, Kevin Bacon graph [65]), and the WWW.

Edge-Reassigning Small-World Network

Watts and Strogatz [60] suggested a probabilistic graph-evolution as the process underlying small-world networks. Evolution starts with a ring lattice with each node connected to its d nearest neighbors. Then each of $\frac{n \cdot d}{2}$ edges is randomly reassigned to distant nodes with a probability p in a round robin fashion. This network has two properties: characteristic-path length L that measures the separation between two nodes (global property), and clustering coefficient C that measures the cliquishness of neighborhood of a node (local property). Clustering is a measure of the extent to which neighbors of a node form a complete graph, and it provides the basis for redefining a Web page in purely

WWW structural terms. The rewired-edges connect nodes that are actually apart from each other by a distance more than $L_{random,}$ where L_{random} is the number of edges in the shortest path, averaged over all pairs of nodes in the network for $p = 1$ (Erdös-Rényi random graph). The chief feature of this model, which can be viewed as yielding random chordal rings if we start with a ring lattice, is that a region of values for p produces evolution unlike the Erdös-Rényi type random-graph evolution. On entering this region from below, the characteristic path-length L of the small-world network decreases dramatically, while the expected clustering is only slowly varying throughout the region. As new edges are reconnected, shortcuts are added between nodes far apart.

An example of the edge-reassigning (Watts-Strogatz) small-world network is shown in Figure 3. The ring lattice has $n = 10$ nodes each with degree $d = 4$, as in Figure 3(a). Each of the 20 edges is reassigned with a probability $p = 0.3$. A small-world network emerges out of the original ring lattice after the six edges are reassigned (Figure 3(b)).

[1] Tendency to form a clique. (See Appendix I)

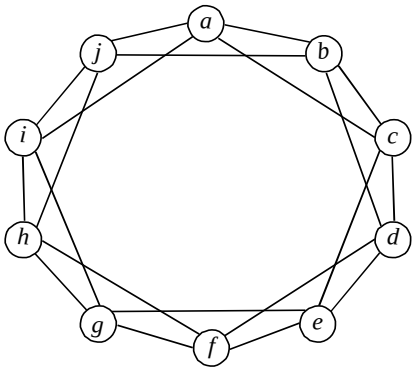


Figure 3(a). A Ring Lattice (n = 10, d = 4)

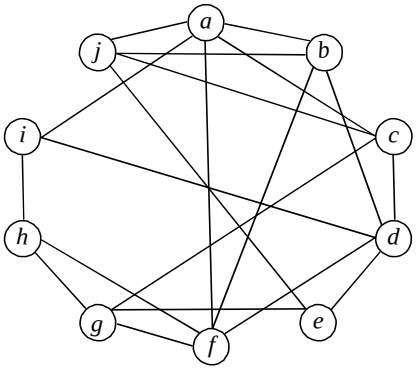


Figure 3(b). Six Edges of Fig. 3(a) Reassigned

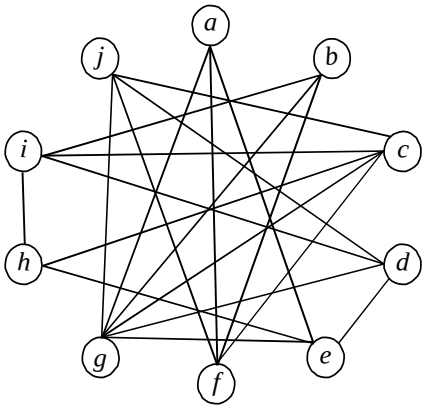


Figure 3(c). All Edges of Fig. 3(a) Reassigned

When $p = 0$, the original regular-graph remains unchanged. The ring lattice is a highly clustered, large world where the characteristic path-length L grows linearly with n. As p increases from 0 to 1, the characteristic path-length L decreases rapidly. However, reassigning an edge from a clustered neighborhood has at most a linear effect on the clustering coefficient C. The result is a poorly clustered, small-world random network. When $p = 1$, we get an Erdös-Rényi type random graph as shown in Figure 3(c).

Edge-Addition Small-World Network

Analysis of the small-world model has gone well beyond the description provided by Watts and Strogatz. Important examples of both results and analytical tools appear in Moore and Newman [50, 51] and Newman *et al.* [53, 54]. The *Edge-Addition* Small-World model extends the Watts-Strogatz small-world model and is based on undirected graphs.

In the edge-addition small-world model, additional edges are added randomly giving an expected number $\frac{p \cdot d \cdot n}{2}$ of new edges. Here *p*, *d*, and *n* denote the probability of addition of a new edge, degree of each node in the original ring-lattice, and the number of nodes in the ring lattice, respectively. The model exhibits the cliquishness and short paths among nodes, found in social networks. An example of edge-addition small-world network is shown in Figure 4. Four edges are added between randomly chosen nodes ($p = 0.2$) in the ring lattice in Figure 3(a).

Moore and Newman [50] consider the following situation in a small-world network. Some fraction *f* of nodes is populated by individuals who will contract a disease if exposed to it. The probability $P(j)$ that a randomly chosen node *i* belongs to a connected cluster of *j* nodes is determined. A cluster has an epidemiological significance. If any node *i* contains an infected individual, $P(j)$ is the probability that *j* people will be conse-

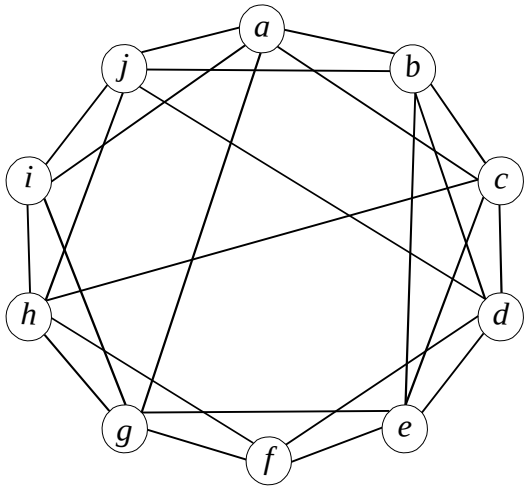


Figure 4. Four Edges Added to Fig. 3(a)

quently infected because there are very short paths from the original infected-individual to all others in the cluster.

The mathematical model presented by Newman *et al.* [53] establishes a connection between the discrete small-world network and its continuous version. It is a mean-field model, meaning that the variables are all taken to be expected values. For a given distance l (expressed in terms of number of edges traversed) and a random node v, the expected number of nodes within distance l of v is determined. These nodes break up into clusters and the number of clusters is a random variable. Using the continuous model, good estimates for these variables are obtained, and corresponding estimates for the discrete model can be obtained (with a suitable confidence measure) that reflect granularity. They derive the distribution of path length and an expression for average path length. The study shows that these parameters are scale-free, *i.e.*, independent of the number of nodes in the network. However, this is only true for 'large' networks. The notion of a 'large' network can be made precise in terms of the correspondence between their continuous and discrete models.

The method presented by Moore and Newman [51] permits the calculation of a threshold value of the fraction f at which the expected size j of infectious outbreak diverges. This represents an epidemic. They also derive an asymptotic expression for $P(j)$ in the neighborhood of this threshold value of f.

The spread of epidemics in a population bears an interesting analogy to the spread of viruses on the WWW. The small-world characteristic of the Web aids in spread of a virus epidemic. A file meant for general distribution on a popular site, if infected by a vi-

rus, can become the source of a major outbreak. The Web pages, particularly primary pages, of a popular Web site are vulnerable as these pages are part of a cluster. Once infected, they propagate the virus through their immediate neighbors and cause a global epidemic thereby causing enormous loss of time and money. The outbreak of virus is going to be significantly less severe and can be controlled, if the infected node does not belong to a near clique. The study of small-world network is useful in predicting the nature of a virus outbreak and its effect on the Web.

3.3 The Preferential-Attachment Model

The small-world models cannot be applied directly to the Web because number of nodes (pages) in the Web is variable. These models do not accommodate a birth/death process in which new pages are created. They also do not explain how new links are formed (possibly through editing old pages), and how both links and pages can be deleted. Two new models have been proposed recently to explain some of the empirical findings concerning the overall Web structure, taking on board the reality of the birth/death process. They are the preferential-attachment model proposed by Albert *et al.* [4, 5, 6] and the Web-site growth model proposed by Huberman and Adamic [35].

The preferential-attachment model starts with a small, null graph having a finite number of nodes (n_0).

1. At each successive time step, a new node with a random, but bounded number of outgoing edges (m_0) is added to the network.

2. The probability that an edge is added between a new node and an existing node i is $\frac{d_i}{\sum_j d_j}$, where, d_i is the degree of node i, and the denominator sum runs over all existing nodes. Thus, a newly introduced node is more likely to be adjacent to a node with high degree. This is the preferential-attachment rule. The preferential attachment of newly introduced nodes implies that the nodes that are added at early stages of development are more likely to have high degree.

Figure 5 shows an example of the preferential-attachment model. It starts with a null graph with four (n_0) nodes (Figure 5(a)). In the first step, a new node e with degree three is added (Figure 5(b)). Node f with three new edges is added, according to the preferential-attachment rule, in the second step (Figure 5(c)). The next two Figures 5(d) and 5(e) show the third and fourth steps, respectively.

The preferential-attachment model reproduces one of the most significant empirical findings about the WWW, namely, the probability that a page or a node i has degree d_i is $\frac{A}{(d_i)^c}$, where A is proportional to the square of average degree of the network and c is a constant. The exponent c was empirically found to be about 2.9, and it is independent of the number of edges being added at each time step. Albert *et al.* analytically derived the exponent c of the power law and found it to be 3.

This model does not allow reconnection of existing edges. Also, addition of new edges takes place only when new nodes are added in the system. However, in real life, the new links are added continuously between old nodes as well.

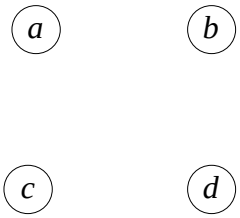


Figure 5(a). Initial Null Graph

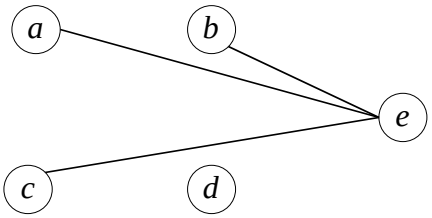


Figure 5(b). Node *e* Added With Degree 3

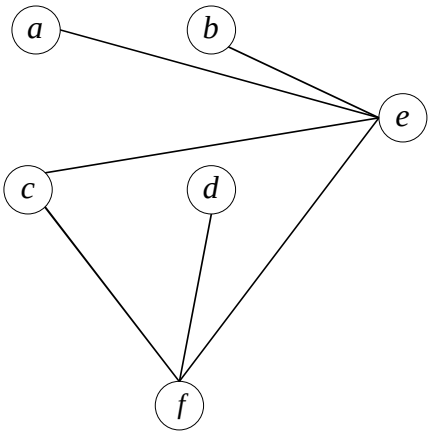


Figure 5(c). Node *f* Added

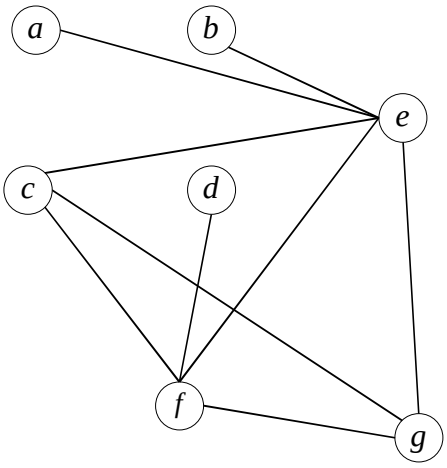


Figure 5(d). Node *g* Added

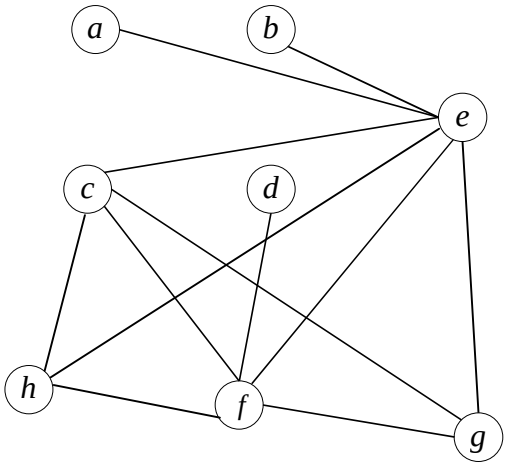


Figure 5(e). Node *h* Added

Figure 5. The Preferential-Attachment Albert *et al.* Model

Albert *et al.* [6] investigated two other models analytically neither of which yield a scale-free power law for node degrees. In one model, the number of nodes increases at each time step, but the preferential-attachment behavior of edge formation between a newly introduced node and an existing node is absent. In the other model, the number of nodes remains fixed; a new edge is added between a randomly selected node and another node with higher degree.

3.4 The Web-Site Growth Model

Huberman and Adamic [1, 35] have proposed another model that exhibits an inverse, scale-free power law for the probability that a Web site s has N_s pages. The scale-free nature means that the power law holds true for any portion of the Web, *i.e.*, it is independent of the number of nodes in the Web subgraph under consideration. The term Web site in their study is defined as a registered domain-name on the Internet. Individual Web pages are arranged in a hierarchical, tree-like manner in each Web site. The basis of the Huberman-Adamic Web-site growth model is an equation governing the number of pages (a random variable) at a site s as a function of time. The number of pages added to site s at any given time is considered proportional to those already existing on the site. The equation has the form

$$N_s(t + 1) = N_s(t) + g(t + 1)\cdot N_s(t),$$

where, $N_s(t)$ = the number of pages at Web site s at time step t, and

$g(t)$ = the universal growth rate, which is independent of a Web site.

Due to the unpredictable nature of growth of a site, $g(t)$ fluctuates about a positive mean g_0, and it can be expressed as

$$g(t) = g_0 + \xi(t),$$

where, g_0 = the basic, constant growth rate, and

$\xi(t)$ = a Brownian motion variable.

The expected value of $N_s(t)$ is

$$N_s(t) = N_s(0) \cdot e^{(g_0 \cdot t + \upsilon_t)},$$

where, υ_t is a Wiener process such that $\upsilon_t^2 = e^{(var(g) \cdot t)}$ and $var(g)$ is variance of growth rate $g(t)$ of the Web site.

The probability of $N_s(t)$ pages at site s is given by a weighted integral, which eliminates dependence on time t and yields a scale-free inverse power-law $\frac{c}{(N_s)^{\gamma}}$, where c is a constant and exponent γ is in the range [1, ∞]. The probability $P(N_s)$ that a given site with an unknown growth rate has N_s pages is given by the sum over all growth rates g, of the probability that the site has so many pages given g, times the probability that a site's growth rate is g.

$$P(N_s) = \sum_i P(N_s / g_i) \cdot P(g_i).$$

Therefore,

Since each particular growth rate gives rise to a power-law distribution with a specific value of the exponent, the above sum is of the form

$$P(N_S) = \frac{c_1}{(N_S)^{\gamma_1}} + \frac{c_2}{(N_S)^{\gamma_2}} + \cdots + \frac{c_n}{(N_S)^{\gamma_n}}.$$

Thus, the probability $P(N_s)$ follows an inverse power-law with an exponent given by the smallest power present in the series.

Huberman and Adamic studied the Web crawls of *Alexa* and *Infoseek* search engines, covering 259,794 and 525,882 sites respectively. The value of exponent γ was found to be in the range [1.647, 1.853] as the 95% confidence interval for the *Alexa* crawl. For the *Infoseek* crawl, γ was estimated to lie in the range [1.775, 1.909] as the 95% confidence interval. Using the value of γ in the power-law equation, the expected number of pages at any site can be estimated. The probability distribution of the number of pages per site for the two Web crawls is shown in Figure 6.

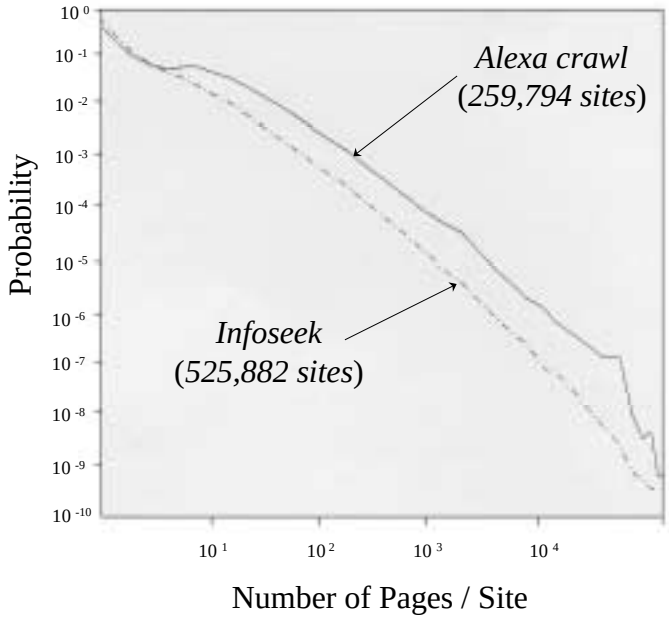


Figure 6. Distribution of Pages per Site
(Courtesy Huberman and Adamic [35])

In a short note, Adamic and Huberman [2] criticized the preferential-attachment model based on its prediction that older pages have larger in-degree than newer pages. Empirical data appears to show no correlation between the age of a page in the WWW and its in-degree. They argue that the growth rate of degree of a page depends on the current degree of a page and not on its age. However, the two models make different assumptions about the creation of new links and the appearance of new pages.

3.5 An Evolving Web-Graph Model

Kumar *et al.* [42, 43] have proposed an evolving random-graph model using two different growth methods. The model evolves over a discrete time step $t = 1, 2, \cdots$. The Web

graph is expressed as $G_t = (V_t, E_t)$ at time t, and the number of nodes at time t, $N_t = |V_t|$. At each time step, new nodes are added given by function $f_v(V_t, t)$ and new edges given by $f_e(f_v, G_t, t)$.

Thus,

$$V_{t+1} = V_t \cup f_v(V_t, t), \qquad \text{and}$$

$$E_{t+1} = E_t \cup f_e(f_v, G_t, t).$$

Two models have been proposed based on the time when a new edge can be connected to a newly formed node.

In the *linear-growth* model, a new node is added at each time step and new edges can be connected to the new node immediately. Therefore, at time step t, a new node u is added and it may be connected to any of the $(t - 1)$ nodes created in earlier steps. Therefore,

$$f_v(V_t, t) = 1, \qquad \text{and} \qquad N_{t+1} = N_t + 1.$$

There is a copy factor $\rho \in (0, 1)$ and a constant out-degree $d^+ \geq 1$ associated with every node. At each time step, a node u is added with d^+ outgoing edges. A prototype node $v \in V_t$ is chosen randomly from existing nodes. For creating the i^{th} outgoing edge from the newly added node u, the destination node is chosen uniformly at random from the existing set of nodes V_t with a probability ρ. With a probability $(1 - \rho)$, node u has an outgoing edge that is copied from the prototype node v. The edge-copying process means that if there is an edge directed from a node v to a node w, then a new edge from the newly added node u to node w is created. The edge-copying Bernoulli process reflects the creation of hyperlinks in a new Web page. Some of the hyperlinks from a new Web page,

on a specific topic, connect the existing Web pages on that topic and remaining hyper-links may connect to Web pages that are not yet recognized for the topic.

An example of the linear-growth model is shown in Figure 7. Initially there is a null graph (Figure 7(a)) with three nodes. The probability ρ is 0.66. Each new node has out-degree of 3. Node *d* is added in Figure 7(b). Node *e* is added with two edges connected to nodes *a* and *b* (Figure 7(c)). The prototype node is selected to be node *d*. As node *d* has an edge directed to node *c*, the third edge from node *e* connects node *c*. Figure 7(d) illustrates the addition of node *f* with two edges connected to randomly-selected nodes *d* and *e*. The prototype node is node *e*. As node *e* has an edge connected to node *b*, the third edge from node *f* has end-node as node *b*.

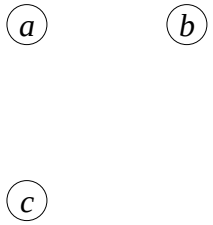


Figure 7 (a). Initial Null Graph

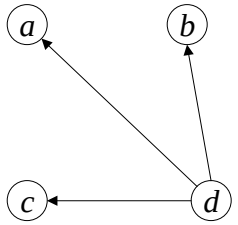


Figure 5(b). Node *d* Added

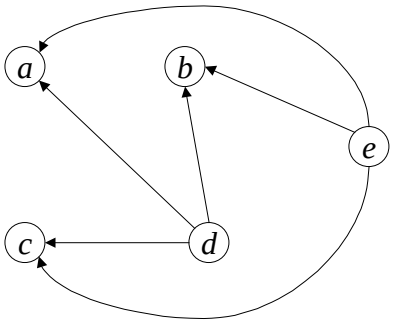


Figure 7(c). Node *e* Added

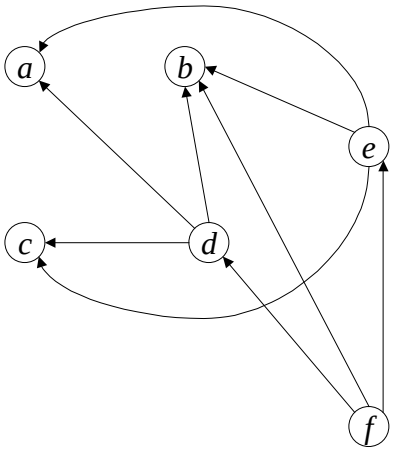


Figure 7(d). Node *f* Added

Figure 7. Evolving Web-Graph Model

The *exponential-growth* model takes into account the fact that newly added Web pages are noticed only after some period. Hence, any node created at time *t* will not have any incoming edge until a certain number of time steps have elapsed. The number of new

nodes created at time step (t + 1) is a constant fraction of the number of nodes at time t. This model has a constant growth factor $g > 0$, self-loop factor $\chi > 1$, tail-copy factor $\chi' \in (0, 1)$, and out-degree factor $\kappa > 0$. The number of nodes added at time step (t + 1) is expressed as a binomial distribution $f_v(V_t, t) \sim B(V_t, g)$. The model assumes $N_1 = 1$ and $N_t = (1 + g)^t$. The number of nodes added at time step (t + 1) is $g \cdot N_t$.

Hence,

$$f_v(V_t, t) = g \cdot N_t , \qquad \text{and} \qquad N_{t+1} = N_t + g \cdot N_t .$$

The new edges at time (t + 1) are created as follows. Each new node is created with χ self-loop edges. For each edge directed to node $u \in V_t$ at time t, a new edge is created with probability $\frac{\kappa \cdot g}{(\kappa + \chi)}$ directed to node u. Assuming that the expected number of edges at step t is $(\kappa + \chi) \cdot N_t$, the number of edges added at step (t + 1) by this process will be $(\kappa \cdot g \cdot N_t)$. The tails of the $(\kappa \cdot g \cdot N_t)$ new edges are chosen according to two different probabilities. The tails of some of the new edges are chosen uniformly at random from among the $g \cdot N_t$ new nodes created during the time step (t + 1), with probability (1 - χ'). The tails of the remaining new edges are chosen at random from among the nodes created in previous steps, with probability χ'; the tail nodes are chosen with probabilities proportional to their current out-degree. Therefore, at time step (t + 1), the number of edges existing is $(\kappa + \chi) \cdot N_{t+1}$, which is $(\kappa + \chi) \cdot g \cdot N_t$.

3.6 The Alpha-Beta Model

Aielo *et al.* [3] have proposed a model for random graphs that follows a scale-free power law for degree. This power-law random-graph model $P(\alpha, \beta)$ has n nodes each with degree d. The degree distribution depends on α and β where α is the logarithm of number of nodes with degree one and β is the log-log rate of decrease for number of nodes with a given degree. Here, n and d satisfy the relationship $\log n = \alpha - \beta \log d$. The model is closer to the conventional random-graph theory and is based on a distribution over all graphs satisfying certain parametric constraints.

4. EMPIRICAL STUDIES

Empirical study is a useful technique for understanding the structure of the Web. Pirolli *et al.* [57] performed the earliest study of link structure of the Web. They studied three kinds of graphs to represent the strength of association among Web pages: (1) Hypertext-link topology, (2) Inter-page text similarity, and (3) Usage paths or flows of users through a locality. By incorporating the usage statistics and page meta-information in these graphs, they studied the Xerox PARC Web server and Xerox's URL *www.xerox.com* to develop techniques for identifying aggregates (clusters) and determining relevancy of hypertext content. Claffy [21] monitored the Internet traffic through network-link infrastructure at a variety of protocol layers. He used a tracking tool called *skitter* to analyze the topology of Internet connection of more than 30,000 sites. Kumar *et al.* [40] studied results of a 1997 crawl from *Alexa, Inc.* covering about 40 million pages. The original data was text-only HTML source and represented the content of over 200 million Web

pages. They pruned the original data by including only the links pointing to Web pages of other sites (*i.e.*, the Web sites different from the link under consideration). In addition, links pointing to pages having identical content were discarded. They found that despite the chaotic nature of content creation on the Web, there exist well-defined communities. These communities can provide reliable and comprehensive information to an interested user. In addition, Web portals can reach their targeted audience effectively by exploiting these Web communities. Albert *et al.* [5] analyzed the induced graph of 325,729-node *nd.edu* and extrapolated the expected distance between any two nodes in the Web graph to be about 19. They also found the inverse power-law characteristic of the degree distribution of a Web page. Huberman and Adamic [35] studied the crawler data comprising 259,794 sites from *Alexa*, and 525,882 sites from *Infoseek* search engine. They found that the probability that a Web site has certain number of Web pages follows inverse-power law. Kleinberg *et al.* [39] report the distribution of bipartite cores on the Web from the result of *Alexa* crawl (using same data as Kumar *et al.* [40]).

4.1 The Bowtie Empirical Study

In one of the most extensive studies, Broder *et al.* [14] analyzed the link structure of a Web subgraph with 203 million pages and 1.5 billion links. Three different sets of experiments were conducted using two *AltaVista* Web crawls.

The first experiment verified the power-law distribution of in-degree (incoming links) of a Web page as reported by Barabási *et al*. [8] and Kumar *et al.* [40]. The exponent of the power law for in-degree was found to be 2.1: the same value as reported by Barabási

et al. [8] and Kumar *et al.* [40]. The out-degree (outgoing links) distribution was also found to exhibit the power law, except in the initial segment.

The second set of experiments explored the connected components of the undirected Web subgraph obtained by ignoring the edge directions. It was found that about 91% of the nodes are connected together and form a weakly connected component (WCC). In addition, the nodes with high in-degree do not affect connectivity of the Web subgraph. These high in-degree nodes are embedded in the graph, which is well connected without

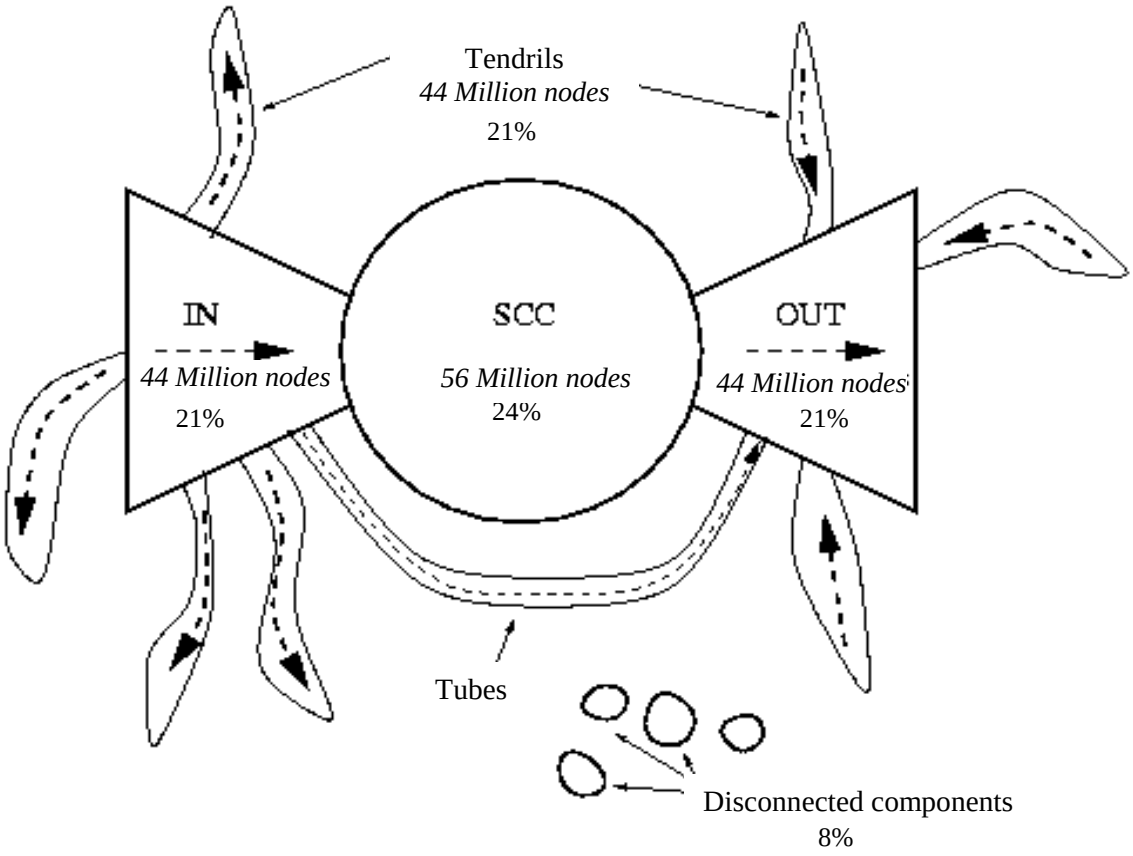


Figure 8. Bowtie map of Web-Page Topology
(Courtesy Broder *et al.* [14])

them. These experiments also analyzed the strongly connected component (SCC) in the directed Web subgraph. There exists a single, large SCC consisting of only 28% (56 million) of total number of Web pages in the experiment.

The third experiment performed Breadth-First Search (BFS) from each of 570 randomly-chosen starting nodes twice: in forward and backward directions. These searches reveal that for some starting nodes, either the forward or the backward BFS traversal covered 100 million nodes and for some starting nodes, both forward and backward traversals covered 100 million nodes. The starting nodes in the latter case lie in the SCC. The nodes in connected component found in the undirected sense (91% of entire sample) form four distinct regions. The first region is the giant SCC. There are directed paths from each node in the SCC component to all other nodes in the SCC. There is a set of newly-formed nodes called IN having only outgoing links and another set of introvert nodes called OUT having only incoming links (*e.g.*, some of the corporate and e-commerce sites). There are directed paths from each node in IN to (all nodes in) SCC and directed paths from (all nodes in) SCC to each node in OUT. There is another set of nodes called TENDRILS, which neither has any directed path going to the SCC nor has any directed path coming from the SCC. There exists a directed path from nodes in IN to TENDRILS and from TENDRILS to nodes in OUT. Each of IN, OUT, and TENDRILS region occupy about 21% of total number of nodes. Finally, some nodes in TENDRILS from the IN region have edges going to nodes in TENDRILS in the OUT region forming a TUBE. A small group of remaining nodes are part of disconnected components, which make up about 8% of the Web. Figure 8 shows the regions of the Web graph that form a bowtie-like structure.

The diameter of the graph studied was found to be greater than 500, and the diameter of the SCC region was estimated to be at least 28. The study shows that only 24% of the time there exists a directed path between any two randomly chosen nodes, and if it exists, the average distance between them is about 16. In addition, ignoring the edge directions (undirected graph), the average distance between any two randomly chosen nodes was found to be 6.83.

4.2 Sampling the Web Through Random Walk

A random walk on a regular, connected, and undirected graph generates a close to uniformly distributed sample of nodes. Therefore, a random walk on the Web can produce an almost uniformly distributed sample of the Web pages. An accurate sampling of the Web helps us to determine the domain-name distribution of Web pages, coverage of search engines, and many important properties of the Web such as average number of links per page and average size of each Web page. Bar-Yossef *et al.* [7] studied random walks on the Web for uniform sampling using the 1996 crawler data of *Alexa*. Their method simultaneously walks the Web and dynamically generates a regular, undirected graph. They conducted walks on the union of SCC and OUT regions (which is connected) described in Section 4.1. However, the graph G formed by the union of SCC and OUT regions is neither regular nor undirected. The edges of this graph G are made bi-directional so that both forward and backward traversal is possible. In addition, number of self-loops are added to each node, so that every node has same degree as that of the node with maximum degree. These modifications make the graph G regular and undirected. The random walk on the connected, regular, and undirected graph, G, can be ab-

stracted as a Markov chain. The mixing time t (number of steps in the walk needed to reach a close to uniform distribution) for an ideal walk is bounded by $O(\frac{1}{\delta}\log_2 n)$ steps,

where, n = total number of nodes in graph G,

δ = eigenvalue gap $|\lambda_1| - |\lambda_2|$,

λ_1 = largest eigenvalue of the transition matrix of the Markov chain, and

λ_2 = second largest eigenvalue of the transition matrix of the Markov chain.

A large eigenvalue gap (δ) indicates that there are a few isolated parts in the graph. Bar-Yossef *et al.* estimated the value of δ to be 10^{-5} for the undirected, regular graph extracted from the 1996 crawler data of *Alexa*. This implies that the mixing time needed for sampling a Web graph with one billion nodes is about three million steps. In the crawler data, only 1 in 30,000 steps of the random walk was not a self-loop and hence, required a hyperlink traversal. Therefore, only 100 Web access is needed for sampling a Web graph with one billion nodes. In the implementation of random walk called *WebWalker*,

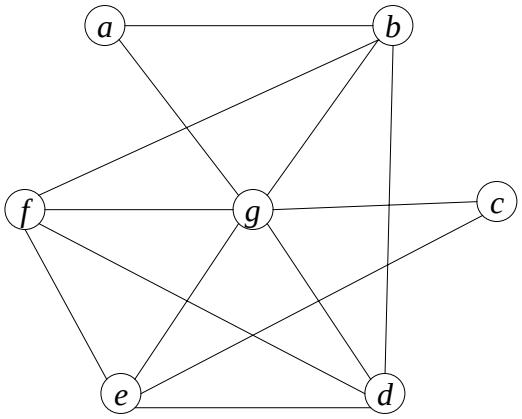


Figure 9(a). A Random Graph

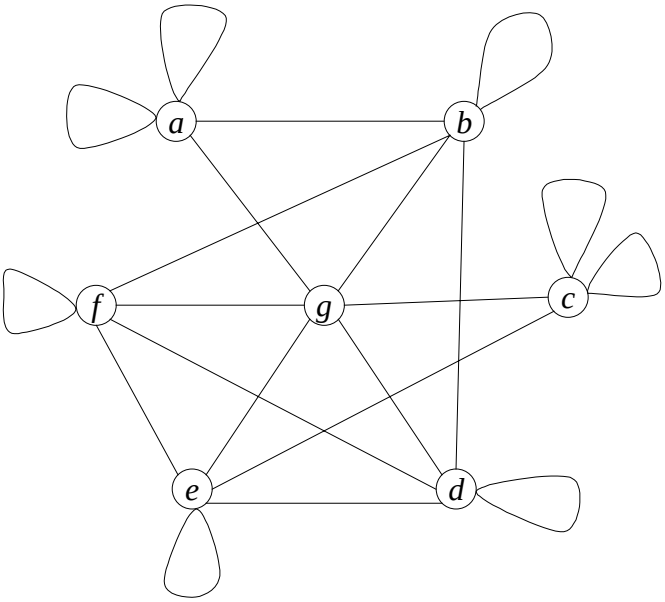


Figure 9(b). Graph Made Regular by Adding Self Loops

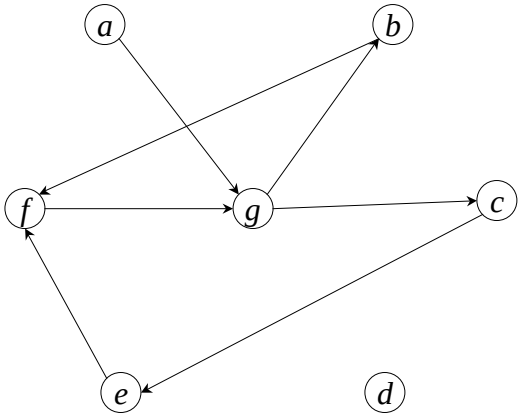


Figure 9(c). An Example of Random Walk

d-regular, undirected graph was built using the resources such as HTML text analysis, search engines, and the random walk itself. In order to determine the domain-name distributions, 21 runs of *WebWalker* with 146,133 hyperlink traversals were performed. It was found that 49.15% of the Web pages were in the .com domain, 8.28% were in the .edu domain, 6.55% were in the .org domain. This domain-name distribution matches a similar study conducted by *Inktomi* [68] in February, 2000. The *WebWalker* study also found that average size of a static HTML page is 11,655 bytes and each page has an average of 9.56 hyperlinks.

Figure 9 shows an example of a random walk performed on a graph. The undirected graph in Figure 9(a) is made regular by addition of self-loops so that each node has degree equal to the degree of node *g* (Figure 9(b)). A random walk starting from node a is illustrated in Figure 9(c).

5. PROPERTIES OF THE WEB GRAPH

Recent empirical and analytical studies of the Web graph have revealed many of its interesting properties. Some of the properties have been established empirically, though there is no theoretical basis discovered until now. We present some important features of the Web graph in this section.

5.1 Power-Law Distribution

Albert *et al.* [4], Broder *et al.* [14], and Kumar *et al.* [40] studied the degree distribution of nodes in the Web graph. They performed empirical studies using graphs of sizes ranging from 325,729 nodes (University of Notre Dame) [4] to 203 million nodes (*AltaVista* crawler data) [14]. It was found that both the in-degree and out-degree of nodes on the Web follow power-law distribution. The number of Web pages having a degree i is proportional to $1/(i^{\varphi})$ where $\varphi > 1$. This implies that the probability of finding a node with a large degree is small yet significant. Both [4] and [14] estimated the exponent of in-degree distribution to be 2.1. According to [4], the out-degree distribution has an exponent of 2.45. The empirical study in [14] shows that the exponent for out-degree distribution is 2.72, though the initial segment of the distribution deviates significantly from

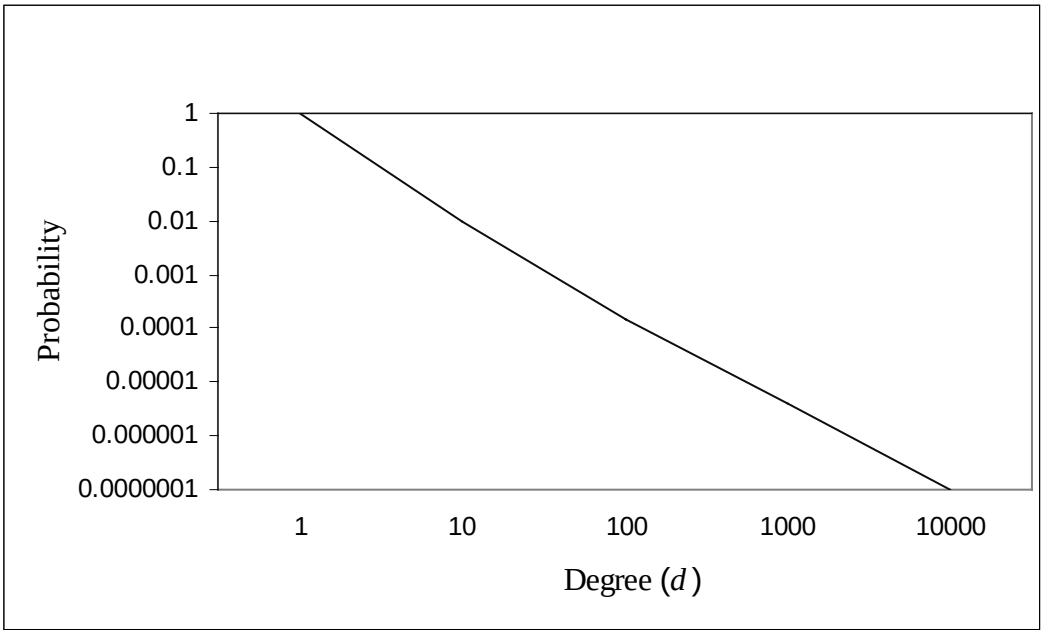


Figure 10. Power-law Distribution of Degree

power law. The average degree of a node in the Web has been found to be seven [39].

5.2 Diameter of the Web Graph

Diameter of the Web graph is important, as it provides an upper bound on the number of clicks needed to reach from one Web page to another. Albert *et al.* [4] calculated the diameter of subgraph induced by the University of Notre Dame site (325,729 nodes and 1,469,680 edges). They consider the diameter as the average distance between any two pair of nodes. It was found that the average diameter of their graph can be expressed as $0.35 + 2.06 \log_{10} n$, where n is number of nodes in the graph. If we were to extend the Notre Dame study to the entire Web graph of two billion nodes, average diameter of the Web would be about 19. Thus the Web, despite its huge size, is a highly connected graph with an average diameter of only 19. The more recent and comprehensive study in [14], discussed in Section 4.1, for a Web subgraph of 203 million nodes and 1.5 billion edges found that the diameter of the SCC portion of the directed Web subgraph was at least 28. The average connected-distance for the directed subgraph was about 16, while the average connected-distance for the same undirected subgraph was found to be 6.83.

5.3 Connected Components

Broder *et al.* [14] studied the size of connected component in the Web through a Web subgraph with 203 million nodes and 1.5 billion edges. They found that 91% of the nodes in the undirected Web subgraph form a weakly connected component (WCC) and their sizes follow a power-law distribution with an exponent of approximately 2.5. The

study of the directed Web subgraph discovered that only 28% of the nodes form a strongly connected component (SCC). The sizes of SCCs also follow the power-law distribution. In their analytical study, Aiello *et al.* [3] investigated the emergence of connected component in random-graph models with power-law distribution.

5.4 Size and Growth Characteristics

Monitoring the continuous growth of the Web reveals many interesting facts. The study in [49] shows that 7.3 million Web pages are being added each day. This study analyzed the *live* growth and acceleration rates of the Web as compared to use of static data by

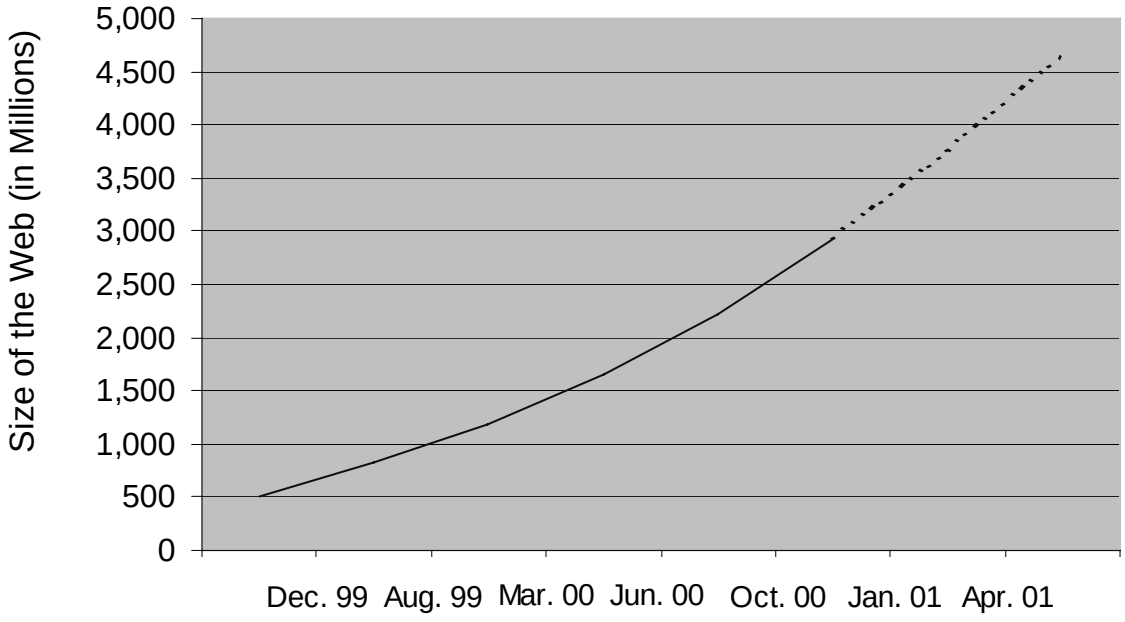


Figure 11. Growth of the Web
(Courtesy Moore and Murray [49])

other studies. The average size of a Web page was found to be 10,060 bytes. On an average, each Web page has 14.38 images on it. As shown in Figure 11, it was predicted that the Web is going to double its size to four billion nodes by February 2001. The expected diameter of the Web increases logarithmically in terms of the size of the Web. If the Web grows from its size of 2.1 billion nodes to 20 billion nodes, the diameter will increase only by 2 from 19.5 to 21.5, respectively. Thus, the diameter grows very slowly with the size of the Web.

5.5 Changing Contents of a Web Page

In a search engine, the index is updated periodically through its crawler. If the crawler knows how often the Web pages change, it may revisit only those page having high probability of change, instead of refreshing the entire search-engine index. A mathematical model for changes to a Web page can aid comparison of different crawling policies. Hence, understanding the lifespan of a Web page can improve the present search-engine technology. Cho and Garcia-Molina [19] studied the change pattern of 720,000 Web pages from 270 Web sites selected from various domains (com, edu, gov, net, org) every day for four months. They found that the average change-interval of a Web page is about four months (approximately). In their study, more than 70% of the Web pages remained unchanged for about one month. It took about 50 days for 50% of the Web pages to change or be replaced by a new page. Changes to a Web page are random events that can be modeled as a Poisson process and this was verified for the Web pages in their sample data.

In another study, to understand how and when the pages change, Brewington and Cybenko [12] studied the contents of over two million Web pages at the rate of 100,000 pages per day for about seven months. In their study, the lifetime of a Web page has been defined as the time between changes made to the page. They modeled the time between modifications of a typical Web page as an exponential distribution characterized by the rate of changes in the page content. This understanding of changes to a Web page is helpful in optimizing the indexing of a search engine. In a (*X*, *Y*)-current search engine having a set of Web pages in its index, a randomly chosen page in the index is current for *Y* time with a probability of at least *X*. The study shows that a (0.95, 1-week)-current search engine must re-index its database at the rate of at least 45 million pages daily.

6. SEARCH ENGINES

The enormous size and rapid growth of the Web makes it difficult for an individual to locate information and navigate by just employing Web addresses. A search engine is useful for locating information in the vast space of the Web. According to *beaucoup.com*, currently there are more than 2,500 search engines (16,240 according to *searchpower.com*), but searching on the Web is still far from perfect.

Bharat and Broder [9] performed one of the earliest studies about the coverage and overlap of search engines. In November 1997, the most comprehensive search engine *AltaVista* covered only 48% of the Web. However, as of March 2000, the coverage of *Altavista* has dropped (Table 1, Section 6.2). Another comparative study of search engines appears in Chu and Rosenthal [20]. The authors compared three major search en-

gines *AltaVista*, *Excite*, and *Lycos* based on their Boolean logic, truncation, field search, word/phrase search, precision, and response time. They found that *AltaVista* offered the best precision results.

In July 1999, Lawrence and Giles [45, 46] found that no search engine indexed more than 16% of the 800-million node Web. Metasearch engines provide a scalable technique to search the ever-growing Web. The metasearch engines send search query to multiple search-engines and show all the results simultaneously. *Inquirus* [44, 47] and *Savvy-Search* [27] are metasearch engines that aim at overcoming the drawbacks of search engines such as low coverage of the Web, inconsistent and inefficient user interfaces, and poor relevance ranking and precision. *Inquirus* downloads the individual Web pages and analyzes them in real time. It is reported to outperform *AltaVista*, *Excite*, *Hotbot*, and many other search engines.

6.1 Architecture of a Search Engine

Search engines consist of three major components: spider, index, and search engine program. The spider or crawler starts with an initial set of URLs called seed URLs, retrieves the Web pages of the seed URLs, and follows the links to other sites from those pages [18, 19]. Keywords found on a Web page are added to the index or catalog of the search engine. The coverage of index, update frequency, and contents of indexed field are important aspects of any search-engine index. The search-engine program finds the relevant pages, from the millions of pages recorded in its index, which match a query and returns them to the user after ranking them in order of relevance.

A search engine (*e.g., google.com*) has distributed crawlers that fetch the Web pages of the URLs sent to it by a URL server [13]. The fetched pages are sent to a store server that compresses and stores the Web page into a repository. A tag is assigned to every Web page in the repository. The indexer of search engine retrieves the compressed Web pages from the repository, uncompresses the Web pages, and parses them. Each page is converted into a set of word occurrences called hits that record the word, position in the Web page, font size, and capitalization. The indexer also parses the links in every Web page and stores information in an anchor file. This anchor file contains information about the end nodes of each link, and the text of the link. A URL resolver converts the relative URLs in anchor file into absolute URLs and generates a database of links. A PageRank is determined for all Web pages in the links database and this PageRank is used to evaluate the relevance of a result (for PageRank algorithm, see Section 7.4.1).

Search engines can increase their speed of indexing Web pages and coverage by optimizing crawler. Chakrabarti *et al.* [17] introduced a goal-directed crawler called *focused crawler* to achieve this. A focused crawler is designed to find the links that are most relevant for the crawl, and avoid the irrelevant regions of the Web, thereby saving hardware, and network resources. The focused crawler selectively finds pages that are relevant to a pre-defined set of topics. This crawler has two components: a classifier which evaluates the relevance of a Web page with respect to the focus topics, and a distiller which identifies nodes that can be good access points to many relevant nodes within few links.

6.2 Issues in the Design of Effective Search-Engines

Ranking the pages returned by a search engine according to relevance is extremely important, especially for queries returning large number of pages. This is done by ranking-function heuristics, which are based on the frequency of occurrence of keywords, and sometimes on the position of keywords in the page. However, such strategies may not deliver correct information. For example, some Web pages may have a keyword repeated many times to attract Web traffic or gain favorable ranking.

One of the factors in the effectiveness of a text-based search engine is the number of Web pages indexed in its database. As shown in Table 1 [66, 67], the search engine with the largest index (*Google*) covers fewer than 35 % of the present Web pages and this disparity is going to increase in future.

Search Engines	**Database Size in Million Pages**
Google	1,346
FAST	575
Webtop	500
Inktomi	500
AltaVista	350
Northern Light	265
Excite	250

Table 1. Index Size of Major Search Engines (as of March 2001)

The keyword searches performed by the current search engines typically turn up large volumes of irrelevant responses. Vast inconsistency in number of hits for two similar queries on a search engine is another serious problem. For example, a query for “Oscar award” on *AltaVista* search engine resulted in 1,480 hits while “award Oscar” yielded

1,208,710 hits. More examples of inefficient search-engine result can be found in Deo *et al*. [26] and Greenlaw *et al*. [32].

7. GRAPH-THEORETIC WEB ALGORITHMS

A higher level of abstraction above a Web page is helpful in understanding the Web topology. In this section, we explore how this abstraction can be used as a tool to identify order and hierarchy in the Web. This understanding is crucial for developing novel search algorithms and enhancing the present search-engine technology. In 1991, Botafogo and Shneiderman [11] were one of the first to apply graph-theoretic techniques for identifying aggregate component among hypertext documents. Their method was based on locating the articulation points in the undirected Web graph and removing them to create a set of subgraphs.

7.1 The WWW Communities

Above the level of a page in the WWW, there are many choices for aggregation of pages. A WWW site has conventionally meant not an IP address, but rather a collection of pages defined by design. Usually this amounts to a topical significance; *e.g.*, pages pointed by some anchor page, often referred to as a home page. In some cases, a site is coherent with respect to some semantical interpretation, *i.e.*, the pages are all about different aspects of one 'thing' but more often the home page is just the hub of the collection.

A WWW site can be also defined through link counting. Here, we consider some notion of locality, *e.g.*, some subnet IP address range and then measure the ratio of the links among all pages inside the range to all the links going outside the range. The threshold of this ratio for assigning pages to a single collection is arbitrary. A more precise formulation can be obtained by computing SCCs. However, in some situations pages related to a main topic might not point outside of themselves, so that the SCC condition could be too stringent to capture many collections of pages that should be identified as sites. Gibson *et al.* [30, 31] have identified two kinds of pages that together make possible a computational concept of a WWW community of pages. This notion is similar to that of a cluster in the Newman-Moore-Watts small-world model. One kind of pages represent authorities

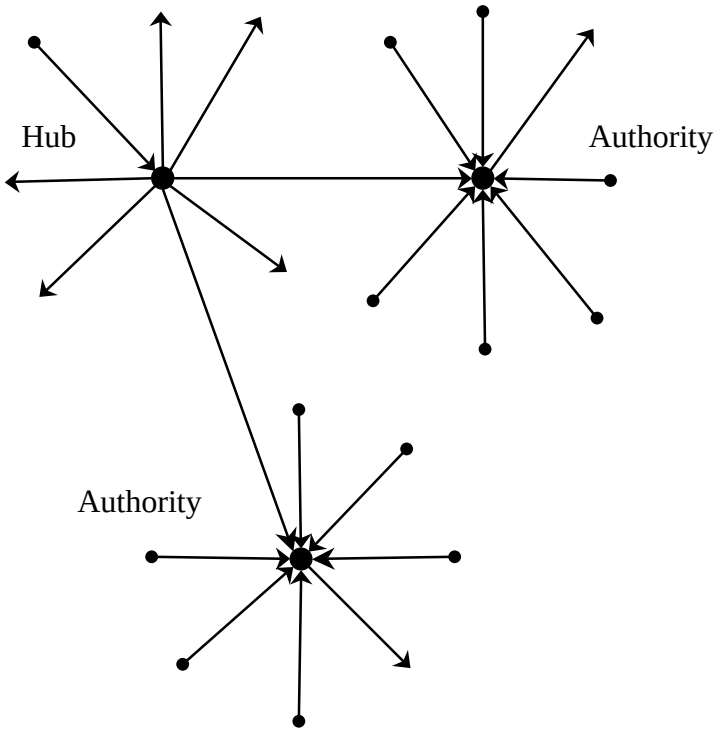


Figure 12. Hubs and Authorities

focusing on a topic while the other kind represent hubs, which point to many authorities.

The abstract community of hubs and authorities arise due to abundance of Web pages that can be returned for any broad-based query. The goal of a search method is to return a small set of "authoritative" pages for the query and this can be achieved by examining the link structure of the Web graph. There is a considerable element of human judgment involved during creation of any hyperlink. These hyperlinks can be exploited for understanding the inherent Web communities. A link from a page *u* to page *v* can be viewed as conferral of authority on page *v*. However, there are many links created which have no meaning with respect to conferral of authority. Counting the incoming links to any Web page is not the complete solution to the problem of identifying authority. This is because there can be pages like *yahoo.com* or *google.com* with very large in-degree, but these Web pages do not form an authority. They are more popular, and hence there can be a popular, but irrelevant Web page.

7.2 HITS Algorithm

Kleinberg [36] proposed an iterative algorithm, called HITS (Hyperlink Induced Topic Search), for identifying authorities and hubs using the adjacency matrix of a subgraph of the WWW. For a broad-topic search, the algorithm starts with a root set *S* of pages returned by a text-based search engine. The set *S* induces a small subgraph focused on the query topic. This induced subgraph is then expanded to include all nodes that are successors of each node in set *S*. In addition, a fixed number of predecessors of each node in the set *S* are also included. Let *G* be the graph induced by the nodes in this expanded node

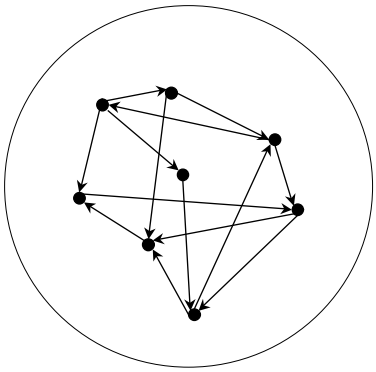


Figure 13(a). Initial Graph *S*

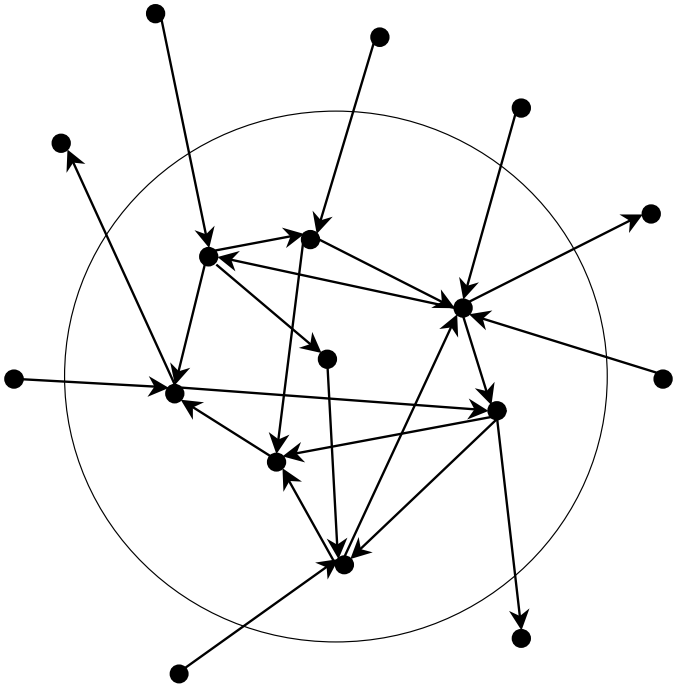


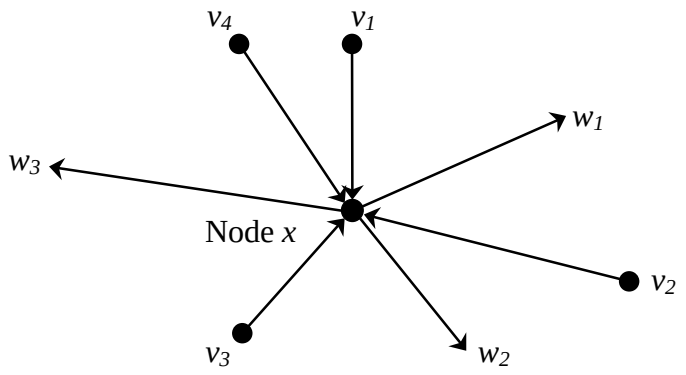
Figure 13(b). Set *S* Expanded to Form set *R*

set. It should be noted that the links that are used purely for navigation within a Web site are not included in this graph *G*. Figures 13(a) and 13(b) illustrate the creation of induced graph *S* and its expansion into graph *G*, respectively.

For each node, *x*, in the graph *G* a non-negative authority weight $a(x)$ and a non-negative hub weight $h(x)$ are computed. The authority and hub weights of all the nodes in graph *G* may be expressed as vectors $a()$ and $h()$, respectively. The elements of vectors $a()$ and $h()$ are initialized to one. In each iteration, $a(x)$ is replaced by the sum of $h(x_i)$'s of all the nodes *i* predecessors to node *x*, and $h(x)$ is replaced by the sum of the $a(x_j)$'s of all the nodes *j* successors of node *x*. The iterations may be expressed as

$$a(x) = \sum_{v \to x} h(v), \quad \text{and} \quad h(x) = \sum_{x \to w} a(w), \quad \text{(see Figure 14).}$$

The authority and hub scores are normalized in each iteration so that $\sum (a(x))^2 = 1$,



$$a(x) = h(v_1) + h(v_2) + h(v_3) + h(v_4), \quad (v_i \in \text{pred}(x))$$
$$h(x) = a(w_1) + a(w_2) + a(w_3), \quad (w_i \in \text{succ}(x))$$

Figure 14. HITS Algorithm

and $\sum (h(x))^2 = 1$. This iterative process converges to yield the authority and hub vector for the initial query. If M is the adjacency matrix of the graph G, then the iterative steps can be viewed as

$$a = M^T \cdot h \,, \qquad \text{and} \qquad h = M \cdot a.$$

This step can be written as

$$a = M^T \cdot h = M^T \cdot M \cdot a \; = (M^T \cdot M) \cdot a, \qquad \text{and}$$

$$h = M \cdot a = M \cdot M^T \cdot a \quad = (M \cdot M^T) \cdot a.$$

Therefore, the iterations of vector a are equivalent to that of multiplying the initial vector a with powers of $M^T M$. These iterations of vector a when normalized, converge to the principal eigenvector of $M^T M$. Similarly, the multiple iterations of normalized vector h converge to the principal eigenvector of MM^T. Thus, HITS applies a link-based computation for identifying the hubs and authorities on a query topic.

7.3 Mining Knowledge-Bases

Communities on the Web can be viewed as forming a bipartite core. A bipartite core $C_{i,j}$ in a graph consists of two (not necessarily disjoint) sets of nodes N_x and N_y such that every node in set N_x has an edge connected to every node in set N_y, where set N_x has i nodes and set N_y has j nodes (*e.g.*, Figure 15). Such bipartite cores form knowledge bases that can be better start points for search and navigation. Kumar *et al.* [41] proposed an elimination/generation algorithm for identifying the core $C_{i,j}$. The algorithm starts with an initial Web subgraph derived from the crawl of a search engine. In the elimination step, nodes whose in-degree or out-degree is a less than a threshold value are pruned from

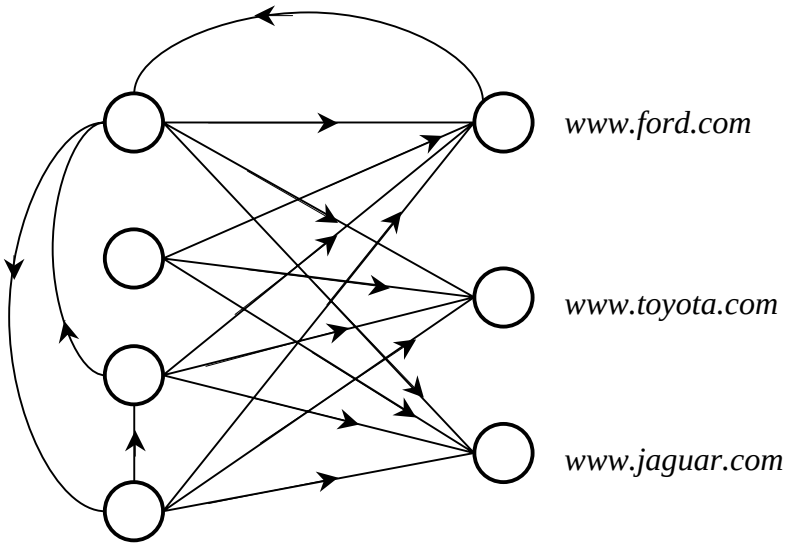


Figure 15. Bipartite Core

initial graph and a residual subgraph is obtained, *e.g.*, for $C_{4,4}$ nodes with in-degree or out-degree smaller than four are eliminated. In the generation step, nodes that barely qualify for inclusion are identified and added to the residual subgraph, *e.g.*, if a node *u* has in-degree exactly four, then it is included in the $C_{4,4}$ core, if and only if the four nodes that point to *u* have a neighborhood of size at least four. The iterative phases of elimination and generation result in a core $C_{i,j}$. Each core is further expanded to form a community by including all the successors of nodes in set N_y and all the nodes that point to at least two nodes in set N_x. The HITS algorithm (Section 7.2) is applied to identify the authorities and hubs for the community. The communities are then indexed according to the most frequent terms used to describe its authority and hub pages. Thus, a knowledge base is built using the graph structure.

7.4 Web-Page Evaluation

A common problem with search engines, that evaluate Web page relevance based on frequency of keywords, is that they can be biased by deliberate inflation of keywords in the Web pages. Another problem, that reduces the likelihood of finding relevant information, is that many Web pages do not contain the keywords that best describe what the Web page is known for or the services it provides. For example, a query for "search engine" at *Altavista* produces none of the major search engines like *Yahoo*, *Lycos*, *Excite*, or *Northern Light* in the top 20 results [36]. Examining *Yahoo* will show that there is nothing on its Web page describing itself as a search engine.

Evaluating the importance of a Web page, using the graph structure of the Web, can solve both these problems. We now present three algorithms for measuring the importance of a Web page.

7.4.1 PageRank Algorithm

Conventional search-engines have relied on matching keywords and strings in the Web pages to index and search the Web for information. *Google*, developed at Stanford University, uses the graph structure of the Web to produce better search results. It uses an algorithm called PageRank [13, 55] that attempts to give a ranking to a Web page, regardless of its content, based solely on its location in the Web graph.

Here,

u = a node (Web page) in the Web graph,

d_i^+ = out-degree of node i,

$w_1, w_2, \ldots, w_k$ = nodes pointing to node u,

η = normalization constant ($\eta < 1$), and

$PR(u)$ = PageRank of a Web page u.

The PageRank of a page u is given as

$$PR(u) = (1-\eta) + \eta \cdot (\frac{PR(w_1)}{d_1^+} + \frac{PR(w_2)}{d_2^+} + \cdots + \frac{PR(w_k)}{d_k^+}).$$

The PageRank algorithm starts with assigning equal rank of one to all pages and recursively computes the PageRank value for each page. The rank of a page is divided equally among its outgoing links. Thus, the PageRank of a page propagates through the link structure of the Web. A page has high PageRank if pages having high PageRank point to it. The PageRank vector $PR()$ corresponds to the principal eigenvector of normalized link-matrix of the Web graph. $PR(u)$ is the probability that a random surfer visits a page u. Normalization constant ,η , is the probability that the random surfer does not follow an

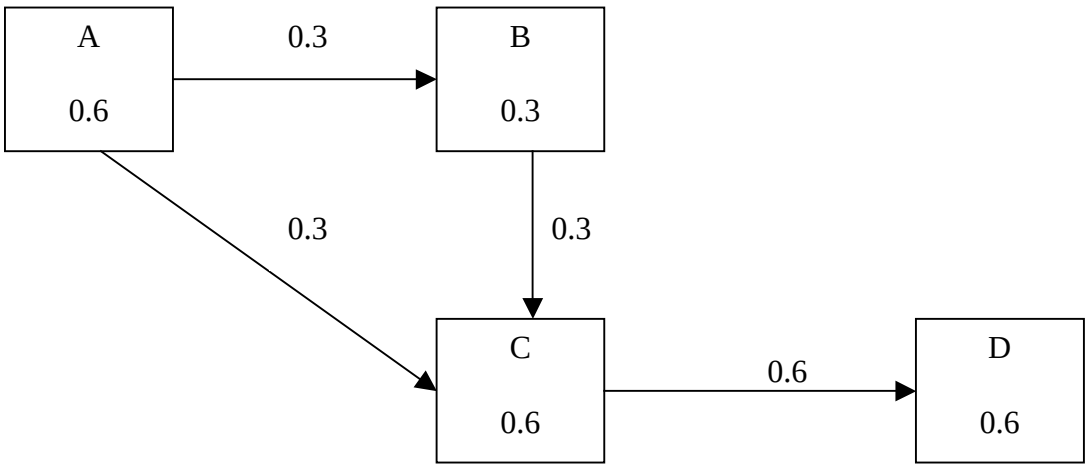


Figure 16. PageRank Computation

outgoing link on page u and selects another page randomly. The quantity η also solves the rank-sink problem caused by the cut nodes in the Web graph. Consider a page *u* having an outgoing link to a page *v*, and pages *v* and *w* have links pointing to each other. If pages *v* and *w* have no outgoing link, then *v* and *w* accumulate rank during the PageRank iteration, without distributing any rank, and a rank sink is created.

An example of the PageRank Computation is shown in Figure 16. The PageRank of page *A* is divided equally amongst its successors *B* and *C*. Since *C* is also a successor of *B*, *C* has effective PageRank of 0.6. Page *D* has PageRank 0.6 as it is a successor of node *C*.

7.4.2 Page Reputation

Computing reputation ranks relies on the model of a random "Web surfer", who is browsing the Web looking for pages relating to a certain topic τ. At each step, the surfer either jumps to a random page that contains the term τ, or follows a random outgoing link from the current page. As this process continues, a reputation value is computed, equal to the number of visits that the random surfer makes to a particular page.

One-Level Reputation Rank Algorithm

This algorithm converges to produce reputation ranks for each page *w* and term τ. Rafiei and Mendelzon [58] provide more detail as well as an algorithm for computing two-level reputation ranks (hub and authority reputation). This algorithm provides a probabilistic formulation for finding the topics that a given Web page is an authority on.

The reputation of a page *w* on topic τ is defined as the probability that a random surfer looking for τ visits page *w*. The notations used in the Rafiei and Mendelzon algorithm are:

N_τ = total number of pages on the Web containing the term τ,

d_x^+ = number of outgoing hyperlinks from page *x*,

p = probability that a random surfer selects a page uniformly at random from a set of pages containing the term τ,

$(1 - p)$ = probability that a random surfer follows an outgoing link from the current page, and

R = a matrix where a row corresponds to a Web page and a column corresponds to each term that appear in the Web page. Each element of the matrix, $R(w, \tau)$, is the reputation value of the Web page *w* with respect to term τ.

The probability that the random surfer visits a page *w* at each step in a random jump is p/N_τ, if page *w* contains term τ and is zero otherwise. If a page *x* is a predecessor of page *w*, then the probability that the surfer visits page *w* at *k* steps after visiting page *x* is $(\frac{(1-p)}{d_x^+})R^{(k-1)}(x,\tau)$ where, $R^{(k-1)}(x,\tau)$ is the probability that the surfer visits page *x* at step $(k - 1)$. The algorithm calculates the probabilities iteratively.

```
For every page w and term τ
    If τ appears in page w
       R(w, τ) = 1/N_τ
    Else R(w, τ) = 0
while R has not converged
    set R'(w, τ) = 0 for every page w and term τ
    For each link x→w
       R'(w, τ) = R'(w, τ) + R(x, τ)/d_x^+
    For every page w and term τ
       R(w, τ) = (1 - p) * R'(x, τ)
    If term τ appears in page w
       R(w, τ) = R(w, τ) + p/N_τ
```

7.4.3 Markov-Chain-Based Rank Method

Zhang and Dong [61] have proposed another ranking algorithm based on the Markov-chain model. The rank function of a Web page (referred as Web resource) is defined as rank: *NV* x *Q*, where *NV* represents a set of Web pages and *Q* represents a set of user queries. *NV* can be viewed as a set of Web pages returned by a search engine. The set of Web pages *NV* and the hyperlinks among its Web pages form a Web subgraph $G = (V, E)$. This algorithm takes into account four parameters: relevance, authority, integrativity, and novelty. The similarity between the contents of a Web page and the user's query q is measured by relevance (ω). The authority of a Web page (μ) is a measure of references

made to the Web page. Integrativity (θ) is a measure of references made in the Web page. This parameter is similar to the hub weight of a page. The novelty metric (ε) measures how a Web page is different from other pages.

While surfing, a user jumps from one page to another and this process can be modeled as a Markov chain. For a query q, $NV = \{nv_1, nv_2, \cdots, nv_n\}$ denotes the set of related Web pages found by the search engine. NV can be viewed as the state space, where a Web page corresponds to a state. For a user surfing the Web at a time t, $p_i(t)$ is the probability that the user is browsing page nv_i and p_{ij} is the probability that the user jumps to another Web page nv_j by following an out-going link from page nv_i. Thus, surfing of the Web can be abstracted as a homogeneous Markov chain.

Consider a random surfer viewing a Web page nv_i at time t. Then at time $(t + 1)$, the random surfer has four choices: continue viewing Web page nv_i, click on a link on Web page nv_i and reach another page, use the "Back" option of browser to return to the previous page, or select another Web page from the results (NV) of the search engine. The tendency matrix takes into account all these four choices available to the user by using relevance (ω), authority (μ), integrativity (θ), and novelty (ε). The tendency matrix W, derived from the graph G, is represented as

$$W_{ij} = \begin{cases} \omega \cdot sim(nv_i, q), & \text{if } i = j \\ \mu, & \text{if } (v_i, v_j) \in E \\ \theta, & \text{if } (v_j, v_i) \in E \\ \varepsilon, & Otherwise. \end{cases}$$

Here, $sim(nv_i, q)$ is the relevance of result nv_i to the query q, $0 < \omega, \mu, \theta, \varepsilon < 1$, and $\omega + \mu + \theta + \varepsilon = 1$.

Normalizing the tendency matrix W results in transition probability matrix T for the set of Web pages NV.

Therefore,

$$T_{ij} = \frac{W_{ij}}{\sum_{j=1}^{n} W_{ij}}, \qquad \text{and} \qquad T = (T_{ij})_{n \times n}.$$

A homogeneous Markov chain's behavior can be determined by its initial distribution vector $T(0)$ and its transition probability matrix T, $T(t) = T(0)\ T^t$. A holomorphic and homogeneous Markov chain, $\{x_t, t \geq 0\}$, with $NV = \{nv_1, nv_2, \cdots, nv_n\}$ as its state space, T as its transition probability matrix, and $T(0)$ as its initial distribution vector, converges to a unique distribution, *i.e.*,

$$\lim_{t \to \infty} T(t) = D.$$

The ultimate distribution vector $D = \{\pi_1, \pi_2, \cdots, \pi_n\}$ is the unique solution of the equation $DT = D$ that satisfies $\pi_i > 0$, $\sum_{i=1}^{n} \pi_i = 1$. This ultimate distribution vector D is the rank of the Web pages. This method calculates the rank of a Web page without any iteration.

7.5 Small-World Algorithmics

Kleinberg [37, 38] proposed an algorithm for finding the shortest or near-shortest path from one node to another in a graph with small expected-diameter. He considered a model of two-dimensional grid with directed edges. Each node in the grid has a directed edge to every other node within a fixed distance *l*, called its local contacts. In addition, each node *u* has a directed edge to Θ other nodes called long-range contacts of node *u*; the i^{th} directed edge from node *u* has end-node *v* with a probability proportional to $[L(u, v)]^{-r}$

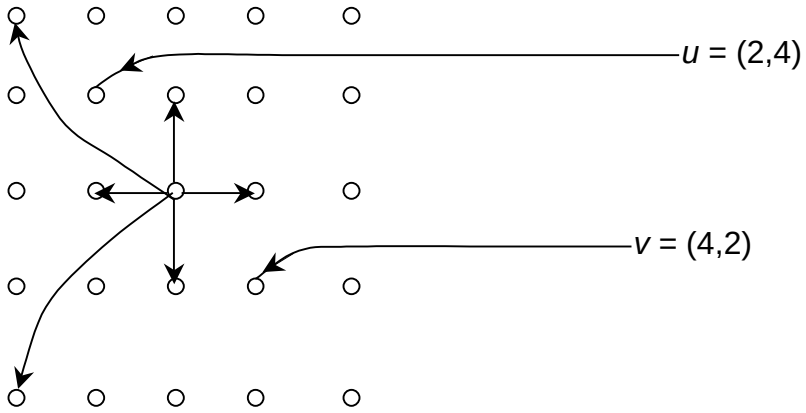


Figure 17. Near-Shortest Path in 2-Dimensional Grid

, ($r \geq 0$), where $L(u, v)$ is the distance between nodes *u* and *v*. Thus, the long-range contacts of a node are clustered in its vicinity by increasing *r*, where *r* is the structural parameter that defines the degree distribution. An example of the two-dimensional grid with $l = 1$ and $\Theta = 2$ is shown in Figure 17. The nodes in the model can be viewed as individuals that have local neighbors as well as a few long-distant acquaintances distrib-

uted geographically. Using an extension of the Watts-Strogatz small-world model, Kleinberg devised a decentralized algorithm that finds a near-shortest path in expected time, polylogarithmic in terms of the size of the graph. The algorithm considers the problem of passing a message from a node *u* to another node *v* using only local information. It assumes that every node knows the location of the target node in the network. In addition, every node knows the locations and long-range contacts of all nodes that have met the message. In each step, an intermediate node *i* passes the message to its adjacent node that is as close to the target node *v* as possible. Kleinberg proved that at $r = 2$, the decentralized algorithm takes advantage of the geographic structure of grid and generates paths having length proportional to $\log n$, where *n* is the number of nodes in the grid. As *r* increases, the long-range contacts of a node become less useful in moving the message. The model can be extended to *k*-dimensional grid, and this decentralized algorithm generates the best polylogarithmic-length path for $r = k$.

7.6 Related-URL and Topic Distillation Algorithms

The graph topology of the Web can be exploited to discover novel search-techniques. Smart Web-agents, that can comprehend the link structure, can be employed to complement the use of search engines. Currently, most search-engine databases are relatively static; as a result, finding most recent information is difficult. Link-based approaches are less susceptible to the keyword-inflating technique applied to influence the search-engine ranking algorithm.

Dean and Henzinger [23] developed a new search paradigm based on hyperlink structure of the Web. They proposed an algorithm for finding Web pages related to a URL. A related Web page is one that addresses the same topic as the original page, but is semantically different. The steps in the Dean-Henzinger algorithm are:

1. Build a vicinity graph for a given URL, *i.e.*, node *U*.

   The vicinity graph is an induced, edge-weighted digraph that includes the URL node *U*, up to *B* randomly selected predecessor nodes of *U*, and for each predecessor node up to B_F successor nodes different from *U*. In addition, the graph includes *F* successor nodes of *U*, and for each successor node up to F_B of its predecessor nodes different from *U*. There is an edge in the vicinity graph if a hyperlink exists from a node *v* to node *w*, provided nodes *v* and *w* do not belong the

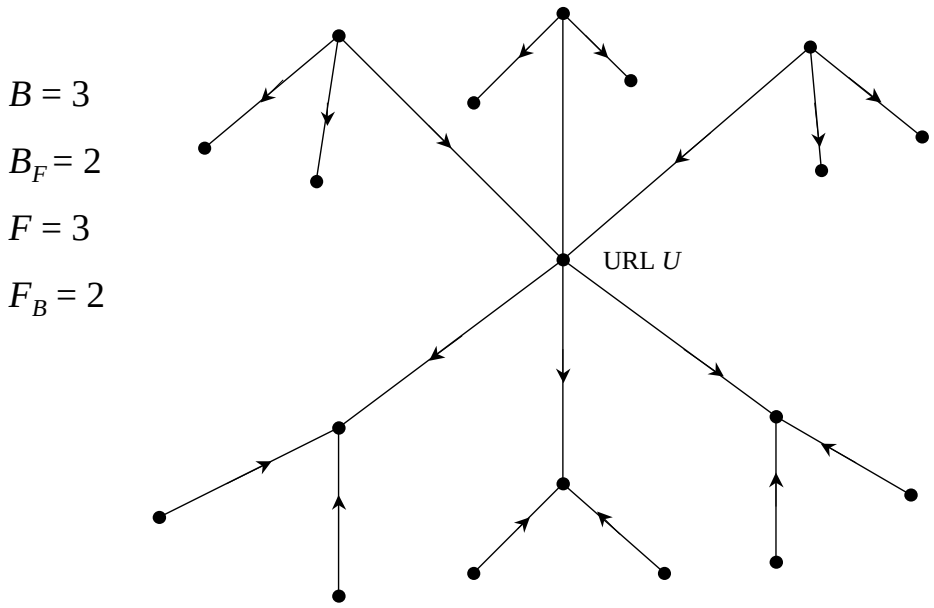


Figure 18(a). Vicinity Graph Formation in Related-URL Algorithm

same Web site. Figure 18(a) illustrates the creation of vicinity graph for $B = 3$, $B_F = 2$, $F = 3$, and $F_B = 3$.

2. Eliminate duplicate and near-duplicate nodes.

   Duplicate nodes are same Web pages on mirror sites or different aliases for same Web page. Two nodes are defined as near-duplicate nodes if they have more than 95% of links in common and each have more than 10 links. The near-duplicate nodes are replaced by a node with links that are union of links of all the near-duplicate nodes.

3. Compute edge weights based on connections between Web sites.

   An edge between nodes on same Web site is assigned a weight 0. If there are m_1 edges directed from a set of nodes on a one Web site to a single node on another Web site, then each edge is given an authority weight $1/m_1$. If there are m_2 edges directed from a single node on a one Web site to a set of nodes on another Web site, each edge is assigned a hub weight $1/m_2$. This prevents the influence of a single Web site on the computation. Figure 18 (b) and 18 (c) illustrate the assigning of edge authority-weight and edge hub-weight to multiple edges from one host to another host.

4. Compute a hub and an authority score for each node in the graph.

   The ten top-ranked authority nodes are returned as the pages that are most related to the start page *U* (modified version of HITS algorithm, Section 7.2).

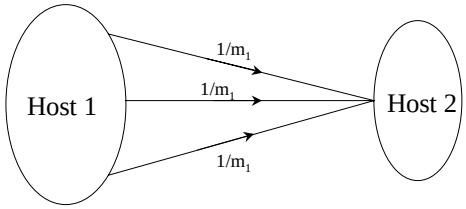


Figure 18(b). Edge-Weight Added for a set of m_1 Edges From Host 1 to Host 2

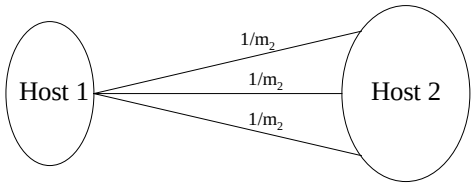


Figure 18(c). Edge-Weight Added for a set of m_1 Edges From Host 1 to Host 2

Bharat and Henzinger [10] proposed a search-engine enhancement algorithm, based on the Web topology, called *Topic distillation*. *Topic distillation* is defined as the process of finding quality Web pages (more relevant) related to a query topic. The topic distillation algorithm solves three problems associated with the HITS algorithm (Section 7.2). The first problem is mutually-reinforced relationship between Web sites (false authority conferral by a set of nodes to another node) where hub and authority score of nodes on each Web site increase. The other two problems are automatically generated links (links carrying no human judgment) and presence of non-relevant nodes. If there are m_1 edges directed from a set of nodes on one Web site to a single node on another Web site, then

rected from a set of nodes on one Web site to a single node on another Web site, then each edge is assigned an authority weight (*edge_auth_wt*) of $1/m_1$. Similarly, if there are m_2 edges directed from a single node on one Web site to a set of nodes on another Web site, then each edge is assigned a hub weight (*edge_hub_wt*) of $1/m_2$. In addition, isolated nodes are eliminated from the graph. The hub weight and authority weight of each node is calculated iteratively as:

$$\forall u \in V,$$
$$a(u) = \sum_{(v,u)\in E} h(v) \times edge\_auth\_wt(v,u), \quad \text{and} \quad h(u) = \sum_{(u,v)\in E} a(v) \times edge\_hub\_wt(u,v).$$

This modification to the HITS algorithm eliminates the mutually-reinforcing relationship problem. The similarity between the query and the node returned by a search engine is defined as the relevance weight of the node. The relevance weight of each node is computed and nodes whose relevance weights fall below a threshold level are eliminated. The elimination step addresses the other two problems.

8. CONCLUSION

The topology of the World Wide Web exhibits the characteristics of a new type of random graph, which at present, is only dimly understood. In this proposal, we have considered several models that help describe the growth of the Web, and we have pointed out some of the features of the Web graph. Recent studies have uncovered only a few fundamental properties of the Web graph. We believe there are still more subtle, but important graph-theoretic properties yet to be discovered about the Web.

Structural analysis of Web connectivity can help its understand the complex regions. This understanding coupled with the information about user traffic traversing the hyper-links can be applied for Web proxy-caching and design of adaptive Web sites. This can also help us design router protocols for prevention of security threats such as denial of services.

The rapid growth of the Web poses a challenge to the present search-engine technology. The solution for improving search quality involves more than just scaling the size of the search-engine index database. Graph-theoretic algorithms, that take into account the link structure of the Web graph, will lead to development of better search engines and smart agents for providing relevant information to the end user, with efficiency and comprehensiveness.

REFERENCES

[1] L. Adamic. The small world web. In *Proceedings of the Third European Conference on Research and Advanced Technology for Digital Libraries.* Paris, France, Sept. 22-24, 1999.
Available at http://www.parc.xerox.com/istl/groups/iea/www/smallworld.html.

[2] L. Adamic and B. Huberman. Technical comment to 'Emergence of scaling in random networks'. *Science*, 287:2115, Mar. 2000.

[3] W. Aielo, F. Chung, and L. Lu. A random graph model for massive graphs. In *Proceedings of the 32nd ACM Symposium on Theory of Computing* (*STOC' 2000*), Portland, OR, pp. 171-180, May 21-23, 2000.

[4] R. Albert, H. Jeong, and A. Barabási. Diameter of the world-wide web. *Nature*, 401:130-131, Sept. 1999.

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

[6] R. Albert, A. Barabási, and H. Jeong. Scale-free characteristics of random networks: The topology of the world wide web. *Physica A*, 281:69-77, 2000.

[7] Z. Bar-Yossef, A. Berg, S. Chien, J. Fakcharoenphol, and D. Weitz. Approximating aggregate queries about web pages via random walks. In *Proceedings of the 26th International Conference on Very Large Databases* (*VLDB' 2000*), Cairo, Egypt, pp. 535-544, Sept. 10-14, 2000.

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

[9] K. Bharat and A. Broder. A technique for measuring the relative size and overlap of public web search engines. In *Proceedings of the Seventh International World Wide Web Conference*, Brisbane, Australia, pp. 379-388, Apr. 14-18, 1998.

[10] K. Bharat and M. Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In *Proceedings of the 21st ACM SIGIR Conference on Research and Development in Information Retrieval*, Melbourne, Australia, pp. 104-111, Aug. 24-28, 1998.

[11] R. A. Botafogo and B. Shneiderman. Identifying aggregates in hypertext structures. In *Proceedings of the Third ACM Conference on Hypertext and Hypermedia*, San Antonio, TX, pp. 63-74, Dec. 15-18, 1991.

[12] B. Brewington and G. Cybenko. How dynamic is the web? In *Proceedings of the Ninth International World Wide Web Conference*, Amsterdam, The Netherlands, May 15-19, 2000.

[13] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the Seventh International World Wide Web Conference*, Brisbane, Australia, Apr. 14-18, 1998.

[14] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the web: experiments and models. In *Proceedings of the Ninth International World Wide Web Conference*, Amsterdam, The Netherlands, May 15-19, 2000.

[15] B. Bollobás. *Random Graphs.* Academic Press, London, 1985.

[16] P. Cao and S. Irani. Cost-aware proxy caching algorithms

[17] S. Chakrabarti, M. Van Den Berg, and B. Dom. Focused crawling: a new approach to topic-specific web resource discovery. In *Proceedings of the Eighth World Wide Web Conference*, Toronto, Canada, May 11-14, 1999.

[18] J. Cho, H. Garcia-Molina, and L. Page. Efficient crawling through URL ordering. In *Proceeedings of the Seventh International World Wide Web Conference*, Brisbane, Australia, Apr. 14-18, 1998.

[19] J. Cho and H. Garcia-Molina. The evolution of the web and implications for an incremental crawler. Stanford Digital Library Technologies Project, Working Paper SIDL-WP-1999-0129, Stanford University, CA, Dec. 1999.

[20] H. Chu and M. Rosenthal. Search engines for the world wide web: A comparative study and evaluation methodology. In *Proceedings of the 59th ASIS annual meeting*, pp. 127-135, Oct. 1996.

[21] K. C. Claffy. Internet measurement and data analysis: topology, workload performance and routing statistics. Technical Report, Cooperative Association for Internet Data Analysis, La Jolla, CA, 1999.
Available at http://www.caida.org/outreach/papers/Nae/.

[22] M. Crovella and A. Bestavros. Explaining world wide web traffic self similarity. Technical Report TR-1995-015, Department of Computer Science, Boston University, Boston, MA, Aug. 1995.

[23] J. Dean and M. Henzinger. Finding related pages in the world wide web. In *Proceedings of the Eighth International World Wide Web Conference*, Toronto, Canada, May 11-14, 1999.

[24] N. Deo. *Graph Theory with Applications to Engineering and Computer Science*. Prentice-Hall, Englewood Cliffs, NJ, 1974.

[25] N. Deo, B. Litow, and P. Gupta. Modeling the Web: Linking Discrete and Continuous. Summer Computer Simulation Conference' 2000. Vancouver, Canada. July 16-20, 2000.

[26] N. Deo, B. Litow, and P. Gupta. Modeling the web: linking discrete and continuous. Technical Report CS-TR-00-003, School of Computer Science, University of Central Florida, Orlando, FL, July 2000.
Available at http://www.cs.ucf.edu/~pgupta/publication.html.

[27] D. Dreilinger and A. Howe. Experiences with selecting search engines using metasearch. *ACM Transactions on Information Systems*, 15(3):195–222, July 1997.

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

[29] D. Florescu, A. Levy, and A. Mendelzon. Database techniques for the world wide web: A survey, *ACM SIGMOD Record*, 27:3, pp. 59-74, Sept. 1998.

[30] D. Gibson, J. Kleinberg, and P. Raghavan. Inferring web communities from link topology. In *Proceedings of the Ninth ACM Conference on Hypertext and Hypermedia*, Pittsburg, PA, pp. 225-234, June 20-24, 1998.

[31] D. Gibson, J. Kleinberg, and P. Raghavan. Structural analysis of the world wide web. *Position paper at the WWW Consortium Web Characterization Workshop*, Cambridge, MA, Nov. 5, 1998.

[32] R. Greenlaw and E. Hepp. *In-line/On-line: Fundamentals of the Internet and the World Wide Web*. McGraw-Hill, New York, NY, pp. 177-181, 1998.

[33] B. Hayes. Graph theory in practice: part I. *American Scientist*, 88(1):9-13, Jan. 2000.

[34] B. Hayes. Graph theory in practice: part II. *American Scientist*, 88(2):104-109, Mar. 2000.

[35] B. Huberman and L. Adamic. Growth dynamics of the world-wide web. *Nature*, 401:131, Sept. 1999.

[36] J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5): 604-632, Sept. 1999.

[37] J. Kleinberg. The small-world phenomenon: an algorithmic perspective. Technical Report 99-1776, Department of Computer Science, Cornell University, Ithaca, NY, Oct. 1999.

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

[39] J. Kleinberg, R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. The web as a graph: measurements, models and methods. *Lecture Notes in Computer Science*, 1627:1-17, July 1999.

[40] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the web for emerging cyber-communities. In *Proceedings of the Eighth International World Wide Web Conference*, Toronto, Canada, May 11-14, 1999.

[41] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Extracting large scale knowledge bases from the web. In *Proceedings of the 25th International conference on Very Large Data Bases* (*VLDB' 99*), Edinburgh, Scotland, Sept. 7-10, 1999.

[42] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal. The web as a graph. In *Proceedings of the 19th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, Dallas, TX, May 15-18, 2000.

[43] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal. Stochastic graph models for the web graph. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, Redondo Beach, CA, Nov. 12-14, 2000.

[44] S. Lawrence and C. Lee Giles. Inquirus, the NECI meta search engine. In *Proceedings* of the *Seventh International World Wide Web Conference*, Brisbane, Australia, pp. 95-105, Apr. 14-18, 1998.

[45] S. Lawrence and C. Lee Giles. Searching the world wide web. *Science*, 280:98-100, Apr. 1998.

[46] S. Lawrence and C. Lee Giles. Text and image metasearch on the web. In *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications* (*PDPTA' 99*), CSREA Press, pp. 829-835, June 28 - July 1, 1999.

[47] S. Lawrence and C. Lee Giles. Accessibility of information on the web. *Nature,* 107:107-109, July 1999.

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

[49] A. Moore and B. H. Murray. Sizing the web. *Cyveilliance, Inc. White Paper*, July 10, 2000.
Available at http://www.cyveillance.com/resources/7921S_Sizing_the_Internet.pdf.

[50] C. Moore and M. E. J. Newman. Epidemics and percolation in small-world networks. *Phy. Rev. E*, 61:5678–5682, 2000.

[51] C. Moore and M. E. J. Newman. Exact solution of site and bond percolation on small-world networks. Working Paper 00-01-007, Santa Fe Institute, Santa Fe, NM, Jan. 2000.

[52] M. E. J. Newman. Models of the small world: a review. Working paper 99-12-080, Santa Fe Institute, Santa Fe, NM, Dec. 1999.

[53] M. E. J. Newman, C. Moore, and D. Watts. Mean-field solution of the small-world network model. Working paper 99-09-066. Santa Fe Institute, Santa Fe, NM, Sept. 1999.

[54] M. E. J. Newman, S. H. Strogatz, and D. J. Watts. Random graphs with arbitrary degree distribution and their applications. Working paper 00-07-042, Santa Fe Institute, Santa Fe, NM, July 2000.

[55] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Stanford Digital Library Project, Working Paper SIDL-WP-1999-0120, Stanford University, CA, 1999.
Available at http://www-diglib.stanford.edu/WP/WWW/WPTitles.html.

[56] J. Pitkow. Summary of WWW characterizations. Technical Report, Xerox PARC, Palo Alto, CA, 1997.

[57] P. Pirolli, P. Pitkow, and R. Rao. Silk from a sow's ear: extracting usable structures from the web. In *Proceedings of the ACM Conference on Human factors in computing* (*CHI' 96*), Vancouver, Canada, pp. 118-125, Apr. 13-18, 1996.

[58] D. Rafiei and A. Mendelzon. What is this page known for? computing web page reputations. In *Proceedings of the Ninth International World Wide Web Conference*, Amsterdam, The Netherlands, May 15-19, 2000.

[59] D. Watts. *Small Worlds: The Dynamics of Networks between Order and Randomness*. Princeton University Press, Princeton, NJ, 1999.

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

[61] D. Zhang and Y. Dong. An efficient algorithm to rank web resources. In *Proceedings of the Ninth World Wide Web Conference*, Amsterdam, The Netherlands, May 15-19, 2000.

[62] http://wombat.doc.ic.ac.uk/foldoc.cgi?query=site&action = search.
FOLDOC: Online Dictionary of Computing.

[63] http://www.netlingo.com/lookup.cfm?term=site.
NetLingo.com: Internet Language Dictionary.

[64] http://webopedia.internet.com/TERM/w/website.html.
Webopedia: Online Computer Dictionary for Internet Terms.

[65] http://www.cs.virginia.edu/oracle/. Oracle of Bacon at Virginia.

[66] http://www.searchenginewatch.com.

[67] http://www.google.com.

[68] htttp://www.inktomi.com.

APPENDIX I — DEFINITIONS, ABBREVIATIONS, AND SYMBOLS

Graph $G = (V, E)$: A graph $G = (V, E)$ consists of a set of objects $V = \{v_1, v_2, \cdots\}$ called nodes, and another set $E = \{e_1, e_2, \cdots\}$ called edges, such that each edge e_k is identified with an unordered pair (v_i, v_j) of nodes.

Directed Graph: A directed graph (or digraph) $G = (V, E)$ consists of a set of nodes $V = \{v_1, v_2, \cdots\}$, a set of edges $E = \{e_1, e_2, \cdots\}$, and a mapping ψ that maps every edge onto some ordered pair of nodes (v_i, v_j).

Null Graph: A graph $G = (V, E)$ having an empty set of edges E is called null graph. All nodes in a null graph have no incident edge and are isolated.

Subgraph: A graph g is said to be a subgraph of a graph G if all the nodes and all the edges of g are in G, and each edge of g has the same end nodes in g as in G.

Induced Subgraph: A subgraph $G' = (V', E')$ of $G = (V, E)$ said to be induced by the node set $V' \subseteq V$ if E' consists of all edges of E whose both end nodes are in V'.

Connected Graph: A graph in which there is at least one path between every pair of nodes is called a connected graph.

Incident: If a node v is an end node of an edge e, v and e are said to be incident on each other.

Degree of a node: The number of edges incident on a node v is called degree, $d(v)$, of node v.

In-Degree: The number of edges incident into a node v is called in-degree $d^-(v)$ of v.

Out-Degree: The number of edges incident out of a node v is called out-degree $d^+(v)$ of v.

Clique: A graph in which there exists an edge between every pair of nodes is called a complete graph or a clique.

Diameter: The diameter of a connected graph is the largest distance between any two nodes in the graph, where distance between the two nodes is defined as the number of edges in the shortest path between the two nodes.

Strongly Connected Component: A directed graph is said to be strongly connected if there is at least one directed path from every node to every other node.

Weakly Connected Component: A directed graph is said to be weakly connected if its corresponding undirected graph is connected, but the directed graph is not strongly connected.

Web Site: A registered domain-name on the Internet is called a Web site (or a host), *e.g.*, www.ucf.edu. Individual Web pages are arranged in a hierarchical, tree-like manner in each Web site.

Abbreviations

BFS	:	Breadth First Search
CAIDA	:	Cooperative Association for Internet Data Analysis
FOLDOC	:	Free Online Dictionary of Computing
HITS	:	Hyperlink Induced Topic Search
HTML	:	Hypertext Markup Language
HTTP	:	Hypertext Transfer Protocol
IP	:	Internet Protocol
SCC	:	Strongly Connected Component
TCP	:	Transmission Control Protocol
URL	:	Uniform Resource Locator
WCC	:	Weakly Connected Component
WWW	:	World Wide Web

List of Symbols

α : Logarithm of number of nodes with degree one

β : Log-log rate of decrease of number of nodes of a given degree

ρ : Copy factor $\in$ (0, 1)

χ' : Tail-copy factor $\in$ (0, 1) in exponential-growth model

χ : Self-loop factor in exponential-growth model ($\chi > 1$)

δ : Eigenvalue gap

σ : Number of edges added to each node in each iteration

κ : Out-degree factor ($\kappa > 0$)

λ_1, λ_2: Largest and second largest eigenvalues of a matrix respectively

φ : Exponent of power-law distribution

$\xi(t)$: Brownian motion variable

γ : Exponent of scale-free inverse power-law and its range is [1, ∞]

η : Normalization constant

τ : Topic of a Web page

ω : Relevance

μ : Authority

θ : Integrativity

ε : Novelty

υ : Weiner process

Θ : Number of long-range contacts (nodes) of a node

$a(x)$: Authority weight for a page x

A : Coefficient proportional to the square of average degree of the network

B : Set of nodes predecessors to a URL node U

B_F : Set of nodes successors to each node in set B

c : Constant

C : Clustering coefficient

$C_{i,j}$: Bipartite core in a graph

d : Degree of a node

d_i^- : In-degree of a node i

d_i^+ : Out-degree of a node i

D : Ultimate distribution vector $D = \{\pi_1, \pi_2, \cdots, \pi_n\}$

edge_auth_wt(u, v): Edge authority-weight of edge (u, v)

edge_hub_wt(u, v): Edge hub-weight of edge (u, v)

E_t : Set of edges in graph $G_t = (V_t, E_t)$

f : Fraction of nodes populated by individuals who will contract a disease

F : Set of nodes successors to URL node U

F_B : Set of nodes predecessors to each node in set F

$g(t)$: Universal growth rate, which is independent of a site

g_0 : Basic, constant growth rate of a Web site

g : Constant growth factor

$h(x)$: Hub weight of a Web page x

G_t : Graph $G_t = (V_t, E_t)$ at time t

i : Randomly chosen node

j : Number of nodes in a cluster

k : Positive integer

$L(u, v)$: Distance between nodes u and v

L_{random} : Length of shortest path averaged over all pair of nodes

L : Characteristic-path length that measures the separation between two nodes

l : Distance between any two nodes in a graph

m : Number of edges in the Web graph

m_0 : Number of edges added in each step of the preferential-attachment model

m_1, m_2 : Number of edges from one Web site to another

M : Adjacency matrix of a graph

M^T : Transpose of matrix M

n : Number of nodes in the Web graph

n_0 : Number of nodes in a null graph

$N_s(t)$: Number of Web pages at site s at time step t

N_x, N_y : Sets of nodes

N_τ : Total number of pages on the Web containing the term τ

NV : Set of Web pages returned by a search engine

p : Uniform probability of edge formation between a pair of nodes

p_c : Critical probability (Erdos-Renyi random graph)

(1 - p) : Probability that a random surfer follows an outgoing link from the current page

$p_i(t)$: Probability that the user is browsing page w_i at time t

p_{ij} : Probability that a user jumps from a Web page w_i to another Web page w_j

pred(x): Set of nodes that are predecessors of node x

$P(j)$: Probability that a randomly chosen node i belongs to a connected cluster of j nodes

$P(N_s)$: Probability that a given site with an unknown growth rate has N_s pages

$PR(u)$: PageRank of a node u

Q : Set of user queries

r : Structural parameter that defines the degree distribution

R : Matrix such that $R(w, \tau)$ is the reputation value of page w with respect to term τ

s : Web site (registered domain name on the Internet)

succ(x): Set of nodes that are successors of node x

S : Root set of Web pages

$sim(N_i, q)$: Relevance of result N_i to the query q

t : Time step

T : Transition probability matrix

u, v, w : Nodes in graph

U : URL (Web address)

x, w : Web page

var(g) : variance of growth rate g of a Web site

V_t : Set of nodes in graph $G_t = (V_t, E_t)$

W : Tendency matrix

(X, Y)-current Search Engine : Search engine in which a randomly chosen page in its index is current for Y time with a probability of at least X

IN, OUT, TENDRILS, TUBES : Regions of the Web graph