The Pennsylvania State University
The Graduate School

DECENTRALIZED ALGORITHMS FOR SEARCH AND ROUTING IN LARGE-SCALE NETWORKS

A Thesis in
Industrial Engineering and Operations Research
by
Hari Prasad Thadakamalla

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

December 2007

The thesis of Hari Prasad Thadakamalla was reviewed and approved* by the following:

Soundar R.T. Kumara
Allen E. Pearce/Allen M. Pearce Professor of Industrial Engineering
Thesis Co-advisor, Co-chair of Committee

Réka Albert
Associate Professor of Physics
Thesis Co-advisor, Co-chair of Committee

Jeya Chandra
Professor of Industrial Engineering
Acting Head of the Department of Industrial Engineering

Ravi Ravindran
Professor of Industrial Engineering

Arvind Rangaswamy
Jonas H. Anchel Professor of Marketing

Leonid Vaserstein
Professor of Mathematics

*Signatures are on file in the Graduate School.

Abstract

During the past decade, advances in technology and science have led to many large-scale distributed systems which can be characterized as networks. Some examples include the World Wide Web, the Internet, the power grid, wireless sensor networks, and military (net-centric) logistics. The scale of the size of these networks is substantially different from the networks considered in traditional graph theory. Further, these networks do not have any pre-specified structure/order or any design principles. Hence, the problems posed in such networks are very novel. Recent years has witnessed an explosion of interest across different disciplines, in understanding and characterizing such large-scale networks, which led to development of a new field called "*Network science*". This activity was mainly triggered by significant findings in real-world networks which led to a revival of network modeling and gave rise to many path breaking results. Until now, a major part of this research was focused on modeling and characterizing the behavior of the networks. However, the ultimate goal of modeling these networks is to understand and optimize the dynamical processes taking place in the network.

Search and routing is one of the most important and prevalent process in many real-world networks. Many times one needs to transport raw material/computer files/messages from one node to another using the edges of the network. In traditional graph theory, there do exist abundant number of algorithms that can compute the optimal paths in the network. However, these algorithms assume that global information of the network is available, i.e. how each and every node is connected in the network is known. But in some scenarios, it is not possible to have access to global information of the network and we need to have decentralized algorithms that can navigate through the network by using only local information. In this dissertation, we address an important process of search and routing in large-scale networks. This forms the core problem of this thesis. Examples include routing of sensor data in wireless sensor networks, locating data

files in peer-to-peer networks, connecting relief workers in a disaster scenario, and finding information in distributed databases. Decentralized search and routing in networks is broadly classified into two types of networks, namely, non-spatial networks and spatial networks.

In non-spatial networks, we study trade-offs presented by local search algorithms in networks which are heterogeneous in edge weights and node degree. We demonstrate that search based on a network measure, local betweenness centrality (LBC), utilizes the heterogeneity of both node degrees and edge weights to perform the best in scale-free weighted networks. We show that the performance of LBC search is similar to BC search, which utilizes the maximum information about a neighbor. Further, we demonstrate that the search based on LBC is universal and performs well in a large class of complex networks. We also test the algorithms on the peer-to-peer network, Gnutella, and discuss the results obtained.

In spatial networks, we consider a family of parameterized spatial network models that are heterogenous in node degree. We investigate several algorithms and illustrate that some of these algorithms exploit the heterogeneity in the network to find short paths by using only local information. In addition, we demonstrate that the spatial network model belongs to a class of *searchable networks* for a wide range of parameter space. Further, we test these algorithms on the U.S. airline network which belongs to this class of networks and demonstrate that searchability is a generic property of the U.S. airline network. These results provide insights on designing the structure of distributed networks that need effective decentralized search algorithms.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Acknowledgments

My years in the graduate school have been unique, conscientious, and filled with enriching experiences. Along with academic knowledge, it has completely transformed my outlook towards life. I would like to express my sincere gratitude to all the people who have been a constant support and guidance during this journey.

My deepest and greatest acknowledgments goes to my family - father Mohan Rao, mother Padma, and sister Harika for their endless patience and constant faith shown in me through out my academic career. I am forever indebted to them for the sacrifices they made to provide me a good education. Much thanks to my extended family members for their blessings, cheerfulness, and well wishes.

I am deeply indebted to my advisors Dr. Kumara and Dr. Albert for their constant guidance, support and encouragement through out my research. I am especially thankful to Dr. Kumara for giving me this opportunity and the freedom to explore novel areas of research. He has been a great mentor and a constant source of encouragement for doing non-conventional research in operations research.

I'm profoundly grateful to my co-advisor Dr. Albert, who has been instrumental in my research. Her remarkable personality, attitude towards research was inspiring and has always fascinated me. It was a great pleasure working with her and I feel fortunate for getting this opportunity.

I thank all the committee members for insightful discussions and critical comments during this research work. I extend my thanks to all the students of LISQ (Laboratory for Intelligent Systems and Quality) who have made the laboratory a convivial place to work. I have spent most of my graduate life in LISQ and it would occupy a special place in my memory for the amalgam of experiences filled with excitement, hard work, and joy. In particular, special thanks to Nandini for being there during the most joyous and most difficult periods in the laboratory. Much thanks to Jorge Cham of PhD comics for adding spice to the graduate lifestyle and raising the spirits of many graduate students. Further, I would like to thank my

undergraduate advisor Dr. Rajendran who has inspired me to go for a Ph.D. I still remember the time when he passionately told me: 'Doing Ph.D. is an incredible experience'. Even though it was not very clear to me at that time, I understand now what he meant.

I'm extremely fortunate to have a great bunch of friends from my childhood to the graduate school. Words couldn't express how much I cherish their friendships and the moments spent with them will last as fond memories. They were always eager participants in doing crazy things that lightened up the tough graduate schedule. The best part is most of our fun activities were done at the spur of the moment. My favorites include: driving all the way to Florida which started of as an half-an-hour driving lesson in Charlottesville, Virginia; late night game club visits to weird places around State College; innumerable number of visits to New York where some include impromptu trips after midnight to taste the hyderabadi biryani or to celebrate Bachu's birthday or Tolani and myself tricking Anu, Malliga, Shak, and Nandini to visit NY. Thanks to Vamsi, Madhukar, Praveen, Nitin, Randhir, Vikranth, Ravi, Rambabu, Rajesh, Nandini, Malliga, Shak, Anu, Vidya, Cicilia, Paddu, Lydia, Masina, Sai, Sanjay, Nazar, Amit, Nanda, and the list goes on... for making my stay at State College pleasant, easy and exciting. Special appreciations to Viswanath, Tolani, Meher, and Bachu who have been there when I needed the most.

Thank you everyone ...

CHAPTER

ONE

Introduction

In the past few years, there have been many path-breaking results in different areas of science and technology, especially in the graph theory [10, 14]. These advances in technology have revolutionized many existing engineering systems and also led to a vast number of possibilities which were not feasible earlier. At the same time, these advances have increased the complexity and scale of the system tremendously which give rise to many new challenges for Operations Research (OR) community [63, 144]. For example, the advances in micro-electro-mechanical systems (MEMS) technology, communications, and processing capabilities have enabled manufacturing tiny and low cost sensors which can sense remote or dangerous locations that were inaccessible earlier [10]. A large number of such tiny sensors which are capable of sensing, communicating and data processing coordinate amongst themselves forming a wireless sensor network (WSN) to achieve a larger sensing task. The sheer number of these tiny sensors and unpredicted dynamics in the network would give rise to many unique challenges in the design of unattended WSNs. Tools and techniques developed in the past are insufficient to deal with the complexity in these systems [122]. We need radically new approaches to address and control many of these new emerging systems. In this dissertation, we try to address these new challenges by utilizing significant advances made during recent years in the new field of "*Network Science*" [128].

Graph theory has been a powerful analytical tool for understanding and solving

various problems in OR. The study on graphs (or networks) traces back to the solution of the Königsberg bridge problem by Euler in 1735. It was the first mathematical proof in graph theory. Later, in the twentieth century, graph theory has developed into a substantial area of study which is applied to solve various problems in engineering and several other disciplines [7]. Euler's great insight lay in representing the Königsberg bridge problem as a graph problem with a set of vertices and edges. Though Euler's representation laid the foundation to graph theory, the size of many networks make them computationally difficult to be analyzed using the traditional exhaustive methods of graph theory. In the last few years there has been an intense amount of activity in understanding and characterizing large-scale complex systems represented as networks, which led to development of a new field called "*Network science*" [128].

1.1 Engineering systems as networks

Many complex engineering systems can be characterized as networks. The individual entities or components can be represented as nodes and interactions between them as edges. For example, sensor networks where sensors can be considered as nodes and connected by an edge if there is a direct communication channel between them. Characterizing them as networks helped researchers to develop various techniques and models in understanding and predicting the behavior of these complex systems [14, 33, 56, 122]. Other examples include:

- *World Wide Web*: It can be viewed as a network where web pages are the nodes and hyperlinks connecting one webpage to another are the directed edges. The World Wide Web is currently the largest network for which topological information is available. It had approximately *one billion* nodes at the end of 1999 [103] and is continuously growing at an exponential rate. A recent study [77] estimated the size to be 11.5 billion nodes as of January 2005.

- *Internet*: The Internet is a network of computers and telecommunication devices connected by wired or wireless links. The topology of the Internet is

studied at two different levels [64]. At the router level, each router is represented as a node and physical connections between them as edges. At the domain level, each domain (autonomous system, Internet Service Provider) is represented as a node and inter-domain connections by edges. The number of nodes, approximately, at the router level were $150,000$ in 2000 [71] and at the domain level were 4000 in 1999 [64].

- *Market graph*: Recently, Boginski *et al.* [34, 35] represented the stock market data as a network where the stocks are nodes and two nodes are connected by an edge if their correlation coefficient calculated over a period of time exceeds certain threshold value. The network had 6556 nodes and $27,885$ edges for the U.S. stock data during the period 2000-2002 [35].

- *Phone call network*: The phone numbers are the nodes and every completed phone call is an edge directed from the receiver to the caller. Abello *et al.* [3] constructed a phone call network from the long distance telephone calls made during a single day which had $53,767,087$ nodes and over 170 million edges.

- *Power grid network*: Generators, transformers, and substations are the nodes and high-voltage transmission lines are the edges. The power grid network of the western United States had 4941 nodes in 1998 [169]. The North American power grid consisted of $14,099$ nodes and $19,657$ edges [12] in 2005.

- *Airline network*: Nodes are the airports and an edge between two airports represent the presence of a direct flight connection [32, 76]. Barthelemy *et al.* [32] have analyzed the International Air Transportation Association database to form the world-wide airport network. The resulting network consisted of 3880 nodes and 18810 edges in 2002.

- *Scientific collaboration networks*: Scientists are represented as nodes and two nodes are connected if the two scientists have written an article together. Newman [119, 120] studied networks constructed from four different databases spanning biomedical research, high-energy physics, computer science and physics. On of these networks formed from Medline database for the period from 1961 to 2001 had $1,520,251$ nodes and $2,163,923$ edges.

- *Movie actor collaboration network*: Another well studied network is the movie actor collaboration network, formed from the Internet Movie Database [1], which contains all the movies and their casts from 1890s. Here again, the actors are represented as nodes and two nodes are connected by an edge if the two actors have performed together in a movie. This is a continuously growing network with $225,226$ nodes and $13,738,786$ edges in 1998 [169].

The above are only a few examples of complex networks pervasive in the real world [14, 33, 56, 122]. The size of these networks is substantially larger from the networks considered in traditional graph theory. Further, these networks do not have any pre-specified structure/order or any design principles. To differentiate these networks from regular graphs they are often called as *complex networks*. These networks are often characterized by diverse behaviors that emerge as a result of non-linear spatio-temporal interactions among a large number of components [144]. Typical behaviors include self-similarity, infinite susceptibility, self-organization, and emergence. The problems posed in such networks are often novel. Tools and techniques developed in the field of traditional graph theory involved studies that looked at networks of tens or hundreds or in extreme cases thousands of nodes and focused on regular graphs. *The substantial growth in size of many such networks (see figure 1.1) and lack of any order in the network necessitates a different approach for analysis and design.* The new methodology applied for analyzing complex networks is similar to the statistical physics approach to complex phenomena.

During the last few years there has been a tremendous amount of research activity dedicated to the study of these complex networks. This activity was mainly triggered by significant findings in real-world networks which are elaborated in chapter 2. There was a revival of network modeling that gave rise to many path breaking results [14, 33, 56, 122] and provoked vivid interest across different disciplines of the scientific community. Prominent models include small-world networks by Watts and Strogatz [169] and scale-free networks by Barabási and Albert [26] and scale-free networks. Until now, as a first step, a major part of this research was focused on modeling and characterizing the behavior of the networks. However, the ultimate goal of modeling these networks is to understand and optimize the dynamical processes taking place in the network. This dissertation focusses on an

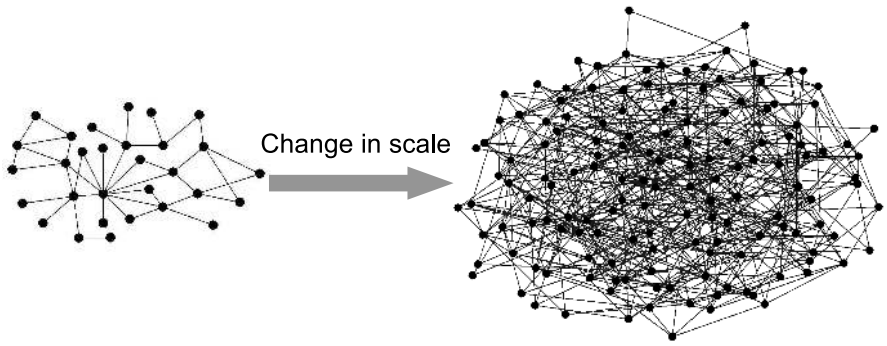


Figure 1.1. Pictorial description of the change in scale in the size of the networks found in many engineering systems. This change in size and lack of any order in the network necessitates a change in the analytical approach.

important process of search and routing in these large-scale networks.

1.2 Search and routing in networks

Search and routing is one of the most important and prevalent process in many real-world networks. In many networks, one needs to route raw material/computer files/messages from one node to another along the edges of the network. Most of the times it is important that the paths used for routing are optimal with respect to resources such as time and cost. Some examples include transporting raw material/finished products from one node to another in supply chain networks; traveling from one place to another using the road network; searching for a person in a social network; routing files from one computer to another in the Internet; searching for a web page on the WWW; traveling from one place to another using the airline network. Finding optimal paths in the networks can be approached in different ways depending upon availability of information. If the information on how each and every node is connected in the network is known, one could use an abundant number of algorithms available in literature for calculating the optimal paths [7, 48]. For instance, one could use breadth first search (BFS) algorithm if all the edges in the network have equal edge weights or use Dijkstra's algorithm if the network has unequal non-negative edge weights. Consider the networks shown in figure 1.2(a) and 1.2(b). The objective is for node 1 to send a message to node

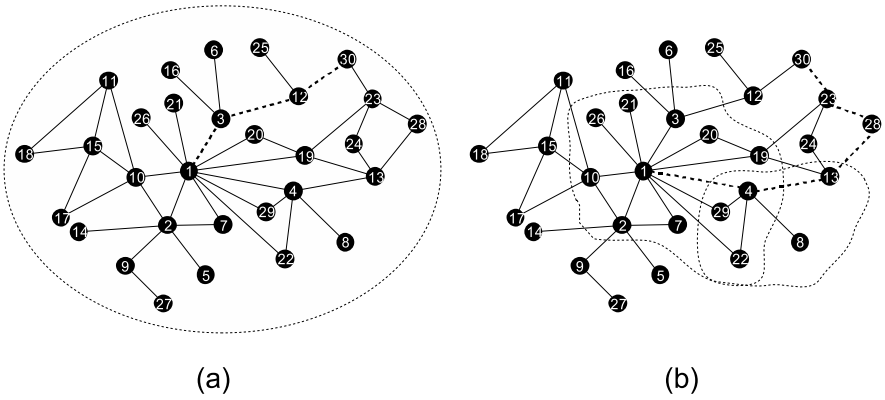


Figure 1.2. Illustration for different ways of routing message from node 1 to node 30. (a) In this case, each node has global connectivity information about the whole network. Hence, node 1 calculates the optimal path and send the message through this path. (b) In this case, each node has only information about its neighbors (as shown by the dotted curve). Using this local information, node 1 tries to send the message to node 30. The path obtained is longer than the optimal path.

30 in the least number of hops. In the network shown in figure 1.2(a), each node has global connectivity information about the network. In such a case, node 1 can calculate the optimal path using traditional algorithms [7] and send the message through this path (1 - 3 - 12 - 30, depicted by the dotted line). However, in some scenarios, it is not possible to have access to the global information of the network and hence need decentralized algorithms that can navigate through the network by using only local information. This forms the core problem of this dissertation.

1.2.1 Problem definition: Decentralized search and routing

Decentralized search and routing is the process in which a node tries to find a network path to a target node using only local information. By local information, we mean that each node has information only about its first, or perhaps second neighbors and it is not aware of nodes at a larger distance and how they are connected in the network. Consider the network shown in figure 1.2 (b), in which each node knows only about its immediate neighbors. Node 1, based on some search algorithm, chooses to send the message to one of its neighbors: in this

case, node 4. Similarly, node 4 also has only local information, and uses the same search algorithm to send the message to node 13. This process continues until the message reaches the target node. We can clearly see that the search path obtained (1 - 4 - 13 - 28 - 23 - 30) is not optimal. However, given that we have only local information available, the research problem in this dissertation is to design optimal search and routing algorithms in different kind of networks. Further, we study how the structure of the network influences the quality of the paths found using local information. The performance of the decentralized algorithms highly depends on the structure of the networks [6, 95, 156, 157, 168]. In some networks, the algorithms can find the paths with lengths in the order of the shortest paths found using global information (the paths with lengths in the order of shortest paths are termed as 'short paths'). These networks which can inherently accommodate local search are called *searchable networks*.

Decentralized search is an intriguing and relatively little studied problem that has many practical applications. In many networks, information such as data files and sensor data is distributed and stored at the nodes of a network. In addition, the nodes have only limited or local information about the network. Examples include routing of sensor data in wireless sensor networks [10, 142], locating data files in peer-to-peer networks [91, 175], and finding information in distributed databases [42]. The importance of search efficiency becomes even more imminent in the case of ad-hoc networks, where the networks are decentralized and distributed, and real time search is required to find the target node. Figure 1.3 provides the pictorial description of the thesis structure and presents the salient points. As shown in this figure, we broadly formulate the decentralized search problem in two types of networks, namely, non-spatial networks and spatial networks. In non-spatial networks, the global position of a node cannot be quantified and it is difficult to know whether a step in the search process is towards the target node or away from the target node. This makes the local search process even more difficult. Whereas in the spatial networks, the global position of the target node can be quantified and each node has this information. This information will guide the search process in reaching the target node quicker.

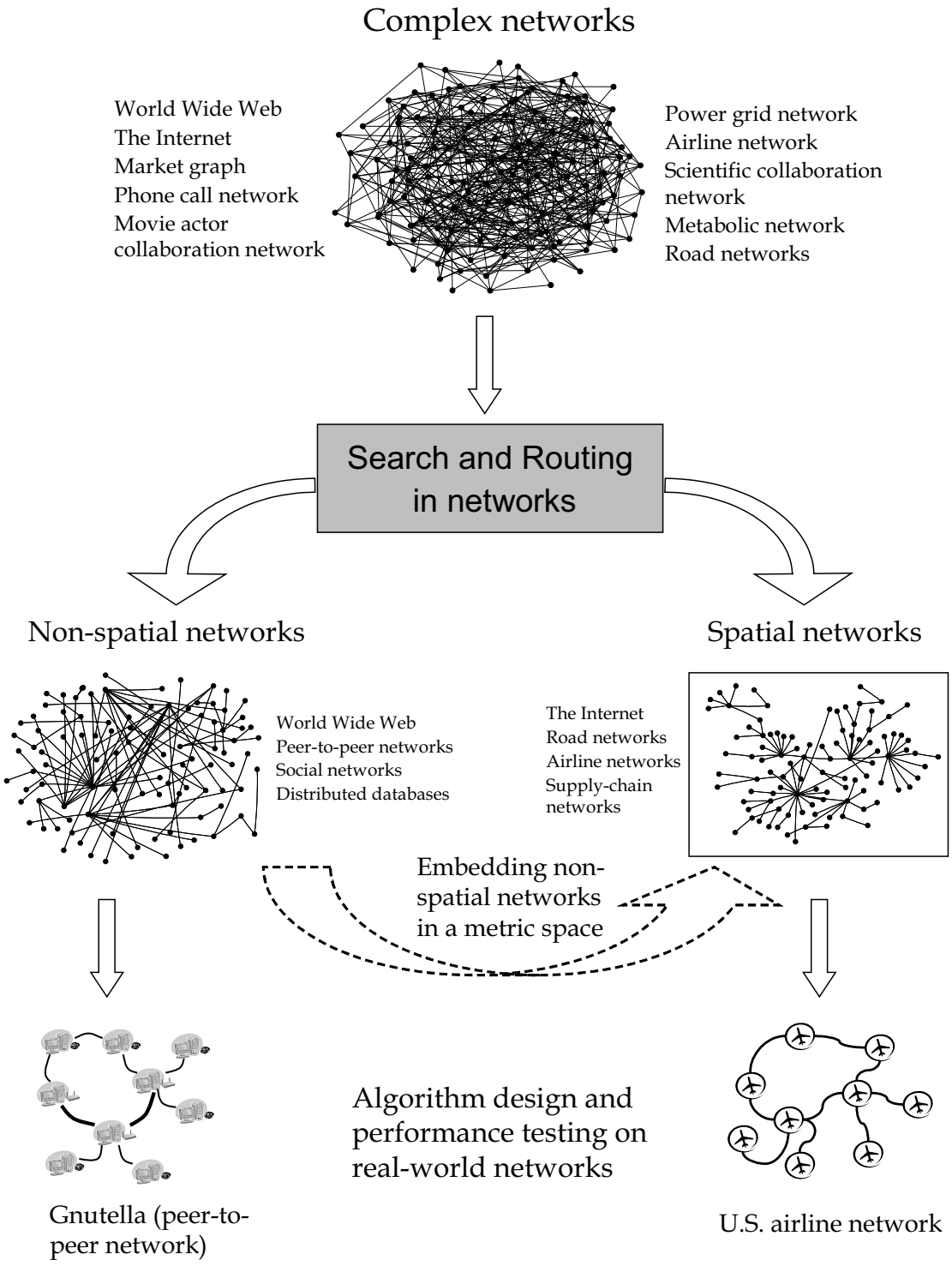


Figure 1.3. Pictorial description of the structure of this dissertation. Firstly, we present an overview of Network Science and then introduce the problem of search and routing in networks. The problem is broadly classified into search in non-spatial networks and search in spatial networks. Later, we present several algorithms for these two problems and test them on real-world networks. We discuss embedding non-spatial networks in a metric space as a part of future work.

1.2.2 Research challenges, objectives, and methodology

Finding short paths in the network using local information alone could be challenging and may be not feasible in many networks. Due to limited information, the algorithms may not be able to choose the edges that lead to the optimal paths. In 1960's, Milgram [111] conducted an experiment to demonstrate that short chains of acquaintances exist between any two people in the social networks. Later, Kleinberg [95] made the even more striking observation that people are able to find these short paths using local information alone. Even though it was demonstrated that short paths can be found using local information, many models in literature do not explain this phenomenon. Kleinberg [95] and later Watts *et al.* [168] argued that the emergence of such a phenomenon requires special topological features. Independently, they proposed different models that explain this phenomenon. Unfortunately, the model proposed by Watts *et al.* is specific to social networks and the model given by Kleinberg represents only a small subset of complex networks. It is not completely clear how the structure of the network influence the performance of the search algorithms. The objective of this thesis is to design and investigate decentralized algorithms in different networks broadly classified as non-spatial and spatial networks.

In non-spatial networks, since the position of the target node cannot be quantified and is unknown, it is extremely difficult to find short paths using local information. In such networks, Adamic *et al.* [6] demonstrated that high-degree search is more efficient than random-walk search, especially, in the networks with power-law degree distribution. In a random-walk search, the node that has the message passes it to a randomly chosen neighbor, and the process continues until it reaches the target node. Whereas, in a high-degree search, the node that has the message passes it to the neighbor with highest degree. However, they assume that the edges in the network are equivalent which does not hold in many real-world networks. In fact, many researchers pointed out that edge-weights have significant influence on many processes in the network [17, 24, 28, 76, 100, 137, 139]. Our objective is to design algorithms that considers the edge-weights and perform better in weighted complex networks. We define a local measure, local betweenness centrality, that considers both the edge-weights and the degree of the neighbors. This measure is adapted from the definition of betweenness centrality [122] and gives the most

central neighbor in the local neighborhood. We consider an algorithm based on this measure and investigate its performance by extensive simulation.

For spatial networks, we consider a family of parameterized spatial network models that are heterogenous in node degree. Many real-world networks such as the Internet [173] and the worldwide airline network [73], can be described by this family of spatial network models. Our objective is to design decentralized search algorithms for this type of network model and demonstrate that this simple model defines a class of searchable networks. We propose several algorithms that consider the heterogeneity in the network and the direction of travel. We investigate their performance for a large range of parameter space in the network model. The following section summarizes the results obtained for these two problems.

1.2.3 Summary of results

For non-spatial networks, we proposed a decentralized search algorithm based on a new local measure called local betweenness centrality. We studied complex trade-offs presented by efficient decentralized search and showed that heterogeneity in edge weights has a huge impact on the search process. Moreover, the impact of edge weights on the performance of the search algorithms increases as the heterogeneity of the edge weights increases. We also demonstrated that the search strategy based on LBC utilizes the heterogeneity in both the node degree and edge weight to perform the best in power-law weighted networks. We observed that the performance of LBC search is similar to BC search, which utilizes the maximum information about a neighbor. Further, we observed that the exponent for the scaling of LBC search with network size decreases as the heterogeneity in edge weights increase. Whereas, the exponent for scaling of high degree search remains the same. This implies that the LBC search exploits low weight edges for navigation. Furthermore, we demonstrated that in unweighted power-law networks, the neighbor with the highest degree is usually the same as the neighbor with the highest LBC. Hence, our proposed search strategy based on LBC is universal and is efficient in a larger class of complex networks. However, when tested in a peer-to-peer network, Gnutella, the results were not consistent with the results obtained from simulations. The reasons for this behavior are not completely clear

and we discuss some possibilities in the future work section.

For spatial networks, we proposed several search algorithms that combine the direction of travel and the degree of the neighbor and illustrated that some of these algorithms exploit the heterogeneity in the network to find short paths by using only local information. In addition, we demonstrated that the spatial network model belongs to a class of *searchable networks* for a wide range of parameter space. Further, we tested these algorithms on the U.S. airline network which belongs to this class of networks and demonstrated that searchability is a generic property of the U.S. airline network. In addition, the spatial network model and the airline network are searchable for a wide range of search algorithms. We demonstrated that direction plays the most important role in efficient search, and even slight blending of direction with degree is sufficient to drastically improve the efficiency of the search algorithms. Hence, searchability is a property of the network rather than of the functional forms used for the search algorithm. As conjectured by others [6, 98], the results presented in this thesis support the hypothesis that many real-world networks evolve to inherently facilitate decentralized search. These results provide insights on designing the structure of distributed networks that need effective decentralized search algorithms.

1.3 Thesis outline

The outline of the thesis is as follows. Chapter 2 introduces the new direction of inter-disciplinary research, *Network Science*, and discusses significant results in the literature. Firstly, we introduce different statistical properties that are prominently used for characterizing large-scale networks. We also present the empirical results obtained for many real complex networks that initiated a revival of network modeling. In section 2.2, we summarize different evolutionary models proposed to explain the properties of real networks. In particular, we discuss Erdős-Rényi random graphs, small-world networks, and scale-free networks. In section 2.3, we discuss the dynamical processes in networks by concentrating on network resilience because of its high relevance to engineering systems and discuss a few other topics briefly. Section 2.4, we discusses the literature on decentralized search in networks which is the primary focus of this thesis.

In chapter 3, we present the problem of decentralized search and routing in large-scale networks. We formulate two research problems and discuss the applications of these problems. In chapter 4, we give the details of the methodology and results obtained for search in non-spatial networks. We present several algorithms and analyze the performance of these algorithms for different types of networks. Further, we present the results obtained for the peer-to-peer network, Gnutella.

In chapter 5, we present the results obtained for the decentralized search problem in spatial networks. We present several algorithms and illustrate that some of these algorithms exploit the heterogeneity in the network to find short paths by using only local information. Further, we test these algorithms on the U.S. airline network which belongs to this class of networks and demonstrate that searchability is a generic property of the U.S. airline network. Finally in chapter 6, we conclude and summarize the results obtained in this thesis. Further, we present the uniqueness of this thesis and discuss potential directions for the future work.

CHAPTER

TWO

Network Science and Optimization - An overview

Many complex distributed systems across different disciples such as communications, sociology, and biology can be characterized as networks, and this in turn allows for understanding their structure, modeling and predicting their behaviors. As discussed in the previous chapter, tools and techniques developed in the field of traditional graph theory focused on regular graphs and involved studies that looked at networks of tens or hundreds or in extreme cases thousands of nodes. For example, consider the problem of finding the shortest route between two geographical points. The problem can be modeled as a shortest path problem on a network, where different geographical points are represented as nodes and they are connected by an edge if there exists a direct path between the two nodes. The weights on the edges represent the distance between the two nodes (see figure 2.1). Let the network be $G(V, E)$ where V is the set of all nodes, E is the set of edges (i, j) connecting the nodes and w is a function such that w_{ij} is the weight of the edge (i, j). The shortest path problem from node s to node t can be formulated as follows.

$$\text{minimize} \sum_{(i,j)\in\xi} w_{ij}x_{ij}$$

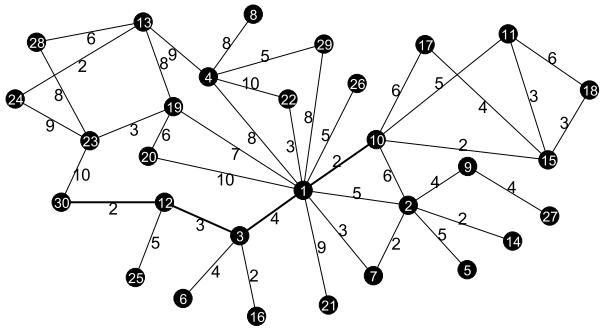


Figure 2.1. Illustration of a typical optimization problem in OR. The objective is to find the shortest path from node 10 to node 30. The values on the edges represent the distance between two nodes. Here we use the exact distances between different nodes to calculate the shortest path 10 - 1 - 3 - 12 - 30.

$$\text{subject to} \sum_{\{j|(i,j)\in\xi\}} x_{ij} - \sum_{\{j|(j,i)\in\xi\}} x_{ji} = \begin{cases} 1 & \text{if } i = s; \\ -1 & \text{if } i = t; \\ 0 & \text{otherwise.} \end{cases}$$

$$x_{ij} \geq 0, \qquad \forall (i,j) \in \xi.$$

where $x_{ij} = 1$ or 0 depending on whether the edge from node i to node j belongs to the optimal path or not respectively. Many algorithms have been proposed to solve the shortest path problem [7]. Using one such popular algorithm (Dijkstra's algorithm [7]), we find the shortest path from node 10 to node 30 as (10 - 1 - 3 - 12 - 30, see figure 2.1). Note that this is a typical instance of the problems solved using the traditional graph theory. However, the problems posed in the complex networks discussed in chapter 1 are very different. This may be due to the size of network or lack of information about the network. The new methodology applied for analyzing complex networks is similar to the *statistical physics* approach to complex phenomena.

The study of large-scale complex systems has always been an active research area in various branches of science, especially in the physical sciences. Some examples are: ferromagnetic properties of materials, statistical description of gases, diffusion, formation of crystals etc. For instance, let us consider a box containing one mole ($6.022 * 10^{23}$) of gas atoms as our system of analysis (see figure 2.2 (a)). If we characterize the system with the microscopic properties of the individual

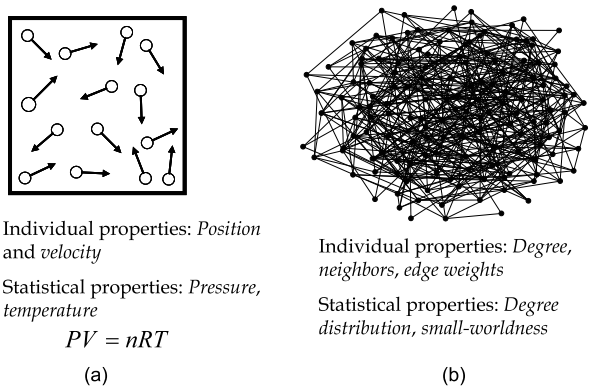


Figure 2.2. Illustration of the analogy between a box of gas atoms and complex networks. (a) A mole of gas atoms ($6.022 * 10^{23}$ atoms) in a box. (b) An example of a large-scale network. For analysis, we need to represent both the systems using statistical properties.

particles such as their position and velocity, then it would be next to impossible to analyze the system. Rather, physicists use statistical mechanics to characterize the system and calculate macroscopic properties such as temperature, pressure etc. Similarly, in networks such as the Internet and WWW, the number of nodes is extremely large and these networks do not have any pre-specified order. Hence, we have to characterize the network using macroscopic properties (such as degree distribution, edge-weight distribution etc), rather than the properties of individual entities in the network (such as the neighbors of a given node, the weights on the edges connecting this node to its neighbors etc) (see figure 2.2 (b)). Now let us consider the shortest path problem in such networks (for instance, WWW). We rarely require specific shortest path solutions such as from node A to node B (from webpage A to webpage B). Rather it is useful if we know the average distance (number of hops) taken from any node to any other node (any webpage to any other webpage) to understand dynamical processes (such as search in WWW). This new approach for understanding networked systems provides new techniques as well as challenges for solving conceptual and practical problems in this field. Furthermore, this approach has become feasible and received a considerable boost by the availability of computers and communication networks which have made the gathering and analysis of large-scale data sets possible.

During the last few years there has been a tremendous amount of research

activity dedicated to the study of these complex networks. This activity was mainly triggered by significant findings in real-world networks which we will elaborate later in this chapter. There was a revival of network modeling which gave rise to many path breaking results [14, 33, 56, 122] and provoked vivid interest across different disciplines of the scientific community. Until now, a major part of this research was focused on modeling and understanding the behavior of the networks. However, the ultimate goal of modeling these networks is to understand and optimize the dynamical processes taking place in the network. In this chapter, we introduce this new direction of inter-disciplinary research (*Network Science*) and discuss significant results in the literature.

2.1 Statistical properties of complex networks

In this section, we explain some of the statistical properties which are prominently used in the literature. These statistical properties help in classifying different kinds of networks. We discuss the definitions and present the empirical findings for many real-world networks.

2.1.1 Average path length and the small-world effect

Let $G(V, E)$ be a network where V is the collection of *entities* (or nodes) and E is the set of *arcs* (or edges) connecting them. A path between two nodes u and v in the network G is a sequence $[u = u_1, u_2, ..., u_n = v]$, where $u_i's$ are the nodes in G and there exists an edge from u_{i-1} to u_i in G for all i. The path length is defined as sum of the weights on the edges along the path. If all the edges are equivalent in the network, then the path length is equal to the number of edges (or hops) along the path. The average path length (l) of a connected network is the average of the shortest paths from each node to every other node in a network. It is given by

$$l \equiv \langle d(u,w) \rangle = \frac{1}{N(N-1)} \sum_{u \in V} \sum_{u \neq w \in V} d(u,w),$$

where, N is the number of nodes in the network and $d(u, w)$ is the shortest path between u and w. Table 5.1 show the values of l for many different networks. Note

Table 2.1. Average path length of many real networks. Note that despite the large size of the network (w.r.t. the number of nodes), the average path length is very small.

Network	Size (number of nodes)	Average path length
WWW [39]	2×10^8	16
Internet, router level [71]	150,000	11
Internet, domain level [64]	4,000	4
Movie actors [169]	212,250	4.54
Electronic circuits [84]	24,097	11.05
Peer-to-peer network [145]	880	4.28

that despite the large size of the network (w.r.t. the number of nodes), the average path length is small. This implies that any node can reach any other node in the network in a relatively small number of steps. This characteristic phenomenon, that most pairs of nodes are connected by a short path through the network, is called the *small-world effect*.

The existence of the small-world effect was first demonstrated by the famous experiment conducted by Stanley Milgram in the 1960s [111] which led to the popular concept of *six degrees of separation*. In this experiment, Milgram randomly selected individuals from Wichita, Kansas and Omaha, Nebraska to pass on a letter to one of their acquaintances by mail. These letters had to finally reach a specific person in Boston, Massachusetts; the name and profession of the target was given to the participants. The participants were asked to send the letter to one of their acquaintances whom they judged to be closer (than themselves) to the target. Anyone who received the letter subsequently would be given the same information and asked to do the same until it reached the target person. Over many trials, the average length of these acquaintance chains for the letters that reached the targeted node was found to be approximately 6. That is, there is an acquaintance path of an average length 6 in the social network of people in the United States. We will discuss another interesting and even more surprising observation from this experiment in section 2.4. Currently, Dodds *et al.* are carrying out an Internet-based study to verify this phenomenon, and initial findings are published in [54].

Mathematically, a network is considered to be small-world if the average path length scales logarithmically or slower with the number of nodes N ($\sim logN$). For example, say the number of nodes in the network, N, increases from 10^3 to 10^6, then

average path length will increase approximately from 3 to 6. This phenomenon has critical implications on the dynamic processes taking place in the network. For example, if we consider the spread of information, computer viruses, or contagious diseases across a network, the small-world phenomenon implies that within a few steps it could spread to a large fraction of most of the real networks.

2.1.2 Clustering coefficient

The clustering coefficient characterizes the local transitivity and order in the neighborhood of a node. It is measured in terms of number of triangles (3-cliques) present in the network. Consider a node i which is connected to k_i other nodes. The number of possible edges between these k_i neighbors that form a triangle is $k_i(k_i-1)/2$. The clustering coefficient of a node i is the ratio of the number of edges E_i that actually exist between these k_i nodes and the total number $k_i(k_i-1)/2$ possible, i.e.

$$C_i = \frac{2E_i}{k_i(k_i-1)}$$

The clustering coefficient of the whole network (C) is then the average of $C_i's$ over all the nodes in the network i.e. $C = \frac{1}{n}\sum_i C_i$ (see figure 2.3). The clustering coefficient is high for many real networks [14, 122]. In other words, in many networks if node A is connected to node B and node C, then there is a high probability that node B and node C are also connected. With respect to social networks, this means that it is highly likely that two friends of a person are also friends, a feature analyzed in detail in the so called *theory of balance* [47].

2.1.3 Degree distribution

The degree of a node is the number of edges incident on it. In a directed network, a node has both an in-degree (number of incoming edges) and an out-degree (number of outgoing edges). The degree distribution of the network is the function p_k, where p_k is the probability that a randomly selected node has degree k. Here again, a directed graph has both in-degree and out-degree distributions. It was found that most of the real networks including the World Wide Web [5, 15, 102], the Internet [64], metabolic networks [87], phone call networks [3, 8], scientific collaboration

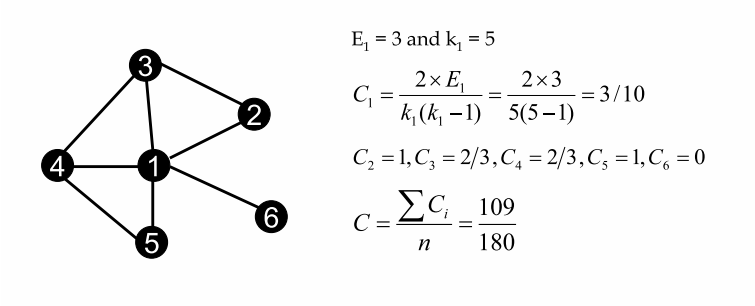


Figure 2.3. Calculating the clustering coefficient of a node and the network. For example, node 1 has degree 5 and the number of edges between the neighbors is 3. Hence, the clustering coefficient for node 1 is 3/10. The clustering coefficient of the entire network is the average of the clustering coefficients at each individual nodes (109/180).

networks [27, 119], and movie actor collaboration networks [13, 20, 26] follow a power-law degree distribution ($p(k) \sim k^{-\gamma}$), indicating that the topology of the network is very heterogeneous, with a high fraction of small-degree nodes and few large degree nodes. These networks having power-law degree distributions are popularly known as *scale-free networks*. These networks were called as scale-free networks because of the lack of a characteristic degree and the broad tail of the degree distribution. Figure 2.4 shows the empirical results for the Internet at the router level and co-authorship network of high-energy physicists. The following are the expected values and variances of the node degree in scale-free networks,

$$E[k] = \begin{cases} finite & \text{if } \gamma > 2; \\ \infty & \text{otherwise.} \end{cases} \qquad V[k] = \begin{cases} finite & \text{if } \gamma > 3; \\ \infty & \text{otherwise.} \end{cases}$$

where γ is the power-law exponent. Note that the variance of the node degree is infinite when $\gamma < 3$ and the mean is infinite when $\gamma < 2$. The power-law exponent (γ) of most of the networks lie between 2.1 and 3.0 which implies that there is high heterogeneity with respect to node degree. This phenomenon in real networks is critical because it was shown that the heterogeneity has a demonstrably large impact on the network properties and processes such as network resilience [12, 16], network navigation, local search [6, 156, 157], and epidemiological processes [132, 133, 134, 135, 136]. Later in this chapter, we will discuss the impact of the this heterogeneity in detail.

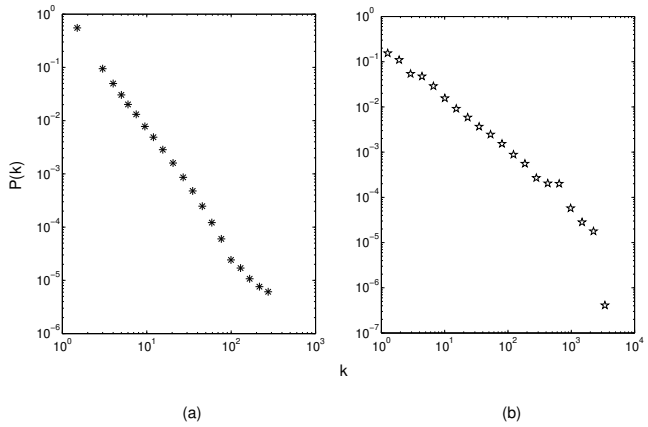


Figure 2.4. The degree distribution of real networks. (a) Internet at the router level. Data courtesy of Ramesh Govindan [71]. (b) Co-authorship network of high-energy physicists, after Newman [119].

2.1.4 Betweenness centrality

Betweenness centrality (BC) of a node is the fraction of shortest paths going through the node. The BC of a node i is given by

$$BC(i) = \sum_{s \neq n \neq t} \frac{\sigma_{st}(i)}{\sigma_{st}},$$

where σ_{st} is the total number of shortest paths from node s to t and $\sigma_{st}(i)$ is the number of these shortest paths passing through node i. If the BC of a node is high, it implies that this node is central and many shortest paths pass through this node. BC was first introduced in the context of social networks [164], and has been recently adopted by Goh *et al.* [69] as a proxy for the load (l_i) at a node i with respect to transport dynamics in a network. For example, consider the transportation of data packets in the Internet along the shortest paths. If many shortest paths pass through a node then the load on that node would be high. Goh *et al.* have shown numerically that the load (or BC) distribution follows a power-law, $P_L(l) \sim l^{-\delta}$ with exponent $\delta \approx 2.2$ and is insensitive to the detail of the scale-free network as long as the degree exponent (γ) lies between 2.1 and 3.0. They further showed that there exists a scaling relation $l \sim k^{(\gamma-1)/(\delta-1)}$ between

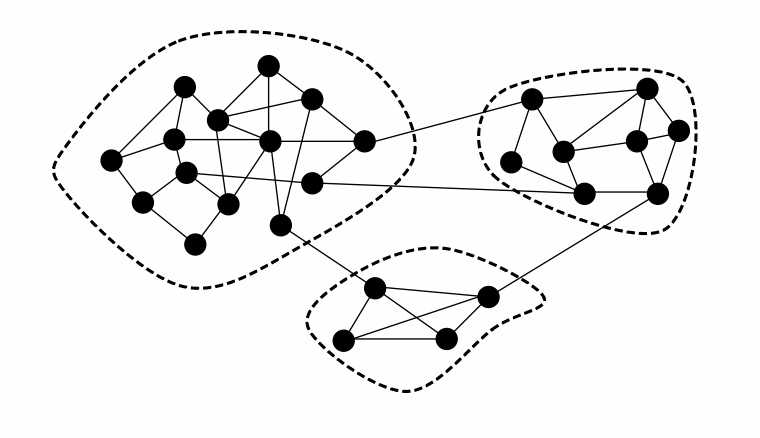


Figure 2.5. Illustration of a network with community structure. Communities are defined as a group of nodes in the network that have higher density of edges within the group than between groups. In the above network, group of nodes enclosed with in a dotted loop is a community.

the load and the degree of a node when $2 < \gamma \leq 3$. Later in this chapter, we discuss how this property can be utilized for local search in complex networks. Many other centrality measures exists in literature and a detailed review of these measures can be found in [99].

2.1.5 Modularity and community structures

Many real networks are found to exhibit a community structure (also called modular structure). That is, groups of nodes in the network have high density of edges within the group and lower density between the groups (see figure 2.5). This property was first proposed in the social networks [164] where people may divide into groups based on interests, age, profession etc. Similar community structures are observed in many networks which reflects the division of nodes into groups based on the node properties [122]. For example, in the WWW it reflects the subject matter or themes of the pages, in citation networks it reflects the area of research, in cellular and metabolic networks it may reflect functional groups [82, 143].

In many ways, community detection is similar to a traditional graph partitioning problem (GPP). In GPP the objective is to divide the nodes of the network into k disjoint sets of specified sizes, such that, the number of edges between these

sets is minimum. This problem is NP-complete [68] and several heuristic methods [80, 93, 140] have been proposed to decrease the computation time. GPP arises in many important engineering problems which include mapping of parallel computations, laying out of circuits (VLSI design) and the ordering of sparse matrix computations [80]. Here, the number of partitions to be made is specified and the size of each partition is restricted. For example, in mapping of parallel computations, the tasks have to be divided between a specified number of processors such that the communication between the processors is minimized and the loads on the processors are balanced. However, in real networks, we do not have any a priori knowledge about the number of communities into which we should divide and about the size of the communities. The goal is to find the naturally existing communities in the real networks rather than dividing the network into a pre-specified number of groups. Since we do not know the exact partitions of a network, it is difficult to evaluate the goodness of a given partition. Moreover, there is no unique definition of a community due to the ambiguity of how dense a group should be to form a community. Many possible definitions exist in literature [66, 124, 130, 141, 164]. A simple definition given in [66, 141] considers a subgraph as a community if each node in the subgraph has more connections within the community than with the rest of the graph. Newman and Girvan [124] have proposed another measure which calculates the fraction of links within the community minus the expected value of the same quantity in a randomized counterpart of the network. The higher this difference, the stronger is the community structure. It is important to note that in spite of this ambiguity, the presence of community structures is a common phenomenon across many real networks. Algorithms for detecting these communities are briefly discussed in section 2.3.2.

2.1.6 Network resilience

The ability of a network to withstand removal of nodes/edges in a network is called network resilience or robustness. In general, the removal of nodes and edges disrupts the paths between nodes and can increase the distances and thus making the communication between nodes harder. In more severe cases, an initially connected network can break down into isolated components that cannot communi-

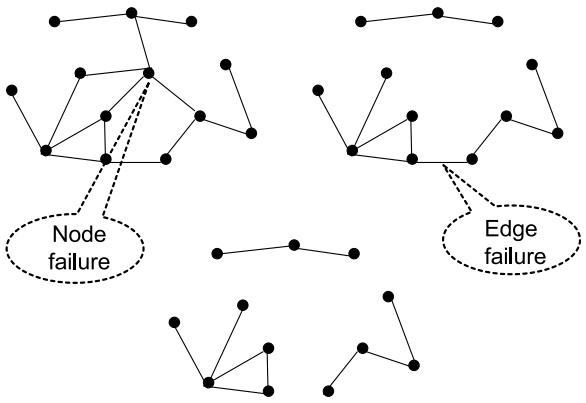


Figure 2.6. Effects of removing a node or an edge in the network. Observe that as we remove more nodes and edges, the network disintegrates into small components/clusters.

cate anymore. Figure 2.6 shows the effect of removal of nodes/edges on a network. Observe that as we remove more nodes and edges, the network disintegrates into many components. There are different ways of removing nodes and edges to test the robustness of a network. For example, one can remove nodes at random with uniform probability or by selectively targeting certain classes of nodes, such as nodes with high degree. Usually, the removal of nodes at random is termed as random failures and the removal of nodes with higher degree is termed as targeted attacks; other removal strategies are discussed in detail in [83]. Similarly there are several ways of measuring the degradation of the network performance after the removal. One simple way to measure it is to calculate the decrease in size of the largest connected component in the network. A connected component is a part of the network in which a path exists between any two nodes in that component and the largest connected component is the largest among the connected components. The lesser the decrease in the size of the largest connected component, the better the robustness of the network. In figure 2.6, the size of the largest connected component decreases from 13 to 9 and then to 5. Another way to measure robustness is to calculate the increase of the average path length in the largest connected component. Malfunctioning of nodes/edges eliminates some existing paths and generally increases the distance between the remaining nodes. Again, the lesser the increase, the better the robustness of the network. We discuss more about network resilience and robustness with respect to optimization in section 2.3.1.

2.2 Modeling of complex networks

In this section, we give a brief summary of different models for complex networks. Most of the modeling efforts focused on understanding the underlying process involved during the network evolution and capture the above-mentioned properties of real networks. In specific, we concentrate on three prominent models, namely, the Erdős-Rényi random graph model, the Watts-Strogatz small-world network model, and the Barabási-Albert scale-free network model.

2.2.1 Random graphs

One of the earliest theoretical models for complex networks was given by Erdős and Rényi [60, 61, 62] in the 1950s and 1960s. They proposed uniform random graphs for modeling complex networks with no obvious pattern or structure. The following is the evolutionary model given by Erdős and Rényi:

- Start with a set of N isolated nodes
- Connect each pair of nodes with a connection probability p

Figure 2.7 illustrates two realizations for Erdős-Rényi random graph model (ER random graphs) for different connection probabilities. Erdős and Rényi have shown that at $p_c \simeq 1/N$, the ER random graph abruptly changes its topology from a loose collection of small clusters to one which has giant connected component. Figure 2.8 shows the change in size of the largest connected component in the network as the value of p increases, for $N = 1000$. We observe that there exists a threshold $p_c = 0.001$ such that when $p < p_c$, the network is composed of small isolated clusters and when $p > p_c$ a giant component suddenly appears. This phenomenon is similar to the *percolation transition*, a topic well-studied both in mathematics and statistical mechanics [14].

In a ER random graph, the mean number of neighbors at a distance (number of hops) d from a node is approximately $< k >^d$, where $< k >$ is the average degree of the network. To cover all the nodes in the network, the distance (l) should be such that $< k >^l \sim N$. Thus, the average path length is given by $l = \frac{\log N}{\log < k >}$, which scales logarithmically with the number of nodes N. This is only an approximate argument for illustration, a rigorous proof can be found in [36].

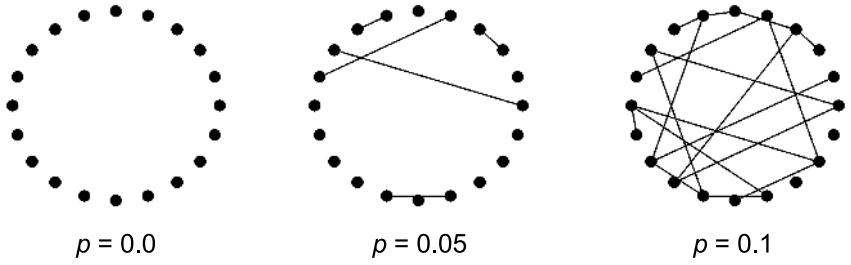


Figure 2.7. An Erdős-Rényi random graph that starts with N = 20 isolated nodes and connects any two nodes with a probability p. As the value of p increases the number of edges in the network increase.

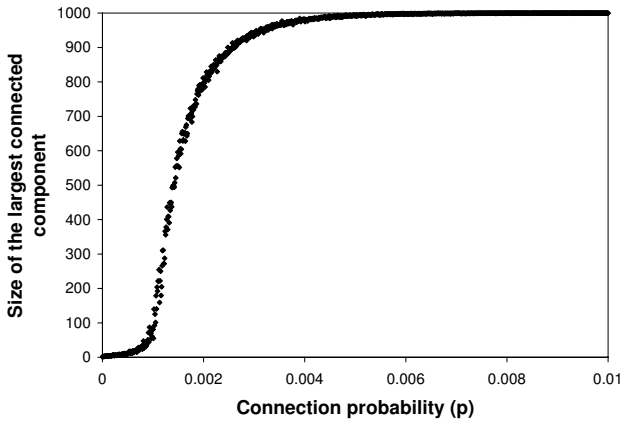


Figure 2.8. Illustration of percolation transition for the size of the largest connected component in Erdős-Rényi random graph model. Note that there exists $p_c = 0.001$ such that when $p < p_c$, the network is composed of small isolated clusters and when $p > p_c$ a giant component suddenly appears.

Hence, ER random graphs are small world. The clustering coefficient of the ER random graphs is found to be low. If we consider a node and its neighbors in a ER random graph then the probability that two of these neighbors are connected is equal to p (the probability that two randomly chosen neighbors are connected). Hence, the clustering coefficient of an ER random graph is $p = \frac{<k>}{N}$ which is small for large sparse networks. Now, let us calculate the degree distribution of the ER

random graphs. The total number of edges in the network is a random variable with an expected value of $pN(N-1)/2$ and the number of edges incident on a node (the node degree) follows a binomial distribution with parameters $N-1$ and p,

$$p(k_i = k) = C_{N-1}^{k} p^k (1-p)^{N-1-k}.$$

This implies that in the limit of large N, the probability that a given node has degree k approaches a Poisson distribution, $p(k) = \frac{<k>^k e^{-<k>}}{k!}$. Hence, ER random graphs are statistically homogenous in node degree as the majority of the nodes have a degree close to the average, and significantly small and large node degrees are exponentially rare.

ER random graphs were used to model complex networks for a long time [36]. The model was intuitive and analytically tractable; moreover the average path length of real networks is close to the average path length of an ER random graph of the same size [14]. However, recent studies on the topologies of diverse large-scale networks found in nature indicated that they have significantly different properties from ER random graphs [14, 33, 56, 122]. It has been found [169] that the average clustering coefficient of real networks is significantly larger than the average clustering coefficient of ER random graphs with the same number of nodes and edges, indicating a far more ordered structure in real networks. Moreover, the degree distribution of many large-scale networks are found to follow a power-law $p(k) \sim k^{-\gamma}$. Figure 2.9 compares two networks with Poisson and power-law degree distributions. We observe that there is a remarkable difference between these networks. The network with Poisson degree distribution is more homogenous in node degree, whereas the network with power-law distribution is highly heterogenous. These discoveries along with others related to the mixing patterns of complex networks [14, 33, 56, 122] initiated a revival of network modeling in the past few years.

Non-uniform random graphs are also studied [8, 9, 44, 112, 117, 125] to mimic the properties of real-world networks, in specific, power-law degree distribution. Typically, these models specify either a degree sequence, which is a set of N values of the degrees k_i of nodes $i = 1, 2, ..., N$ or a degree distribution $p(k)$. If a degree distribution is specified then the sequence is formed by generating N random values from this distribution. This can be thought as giving each node i in the

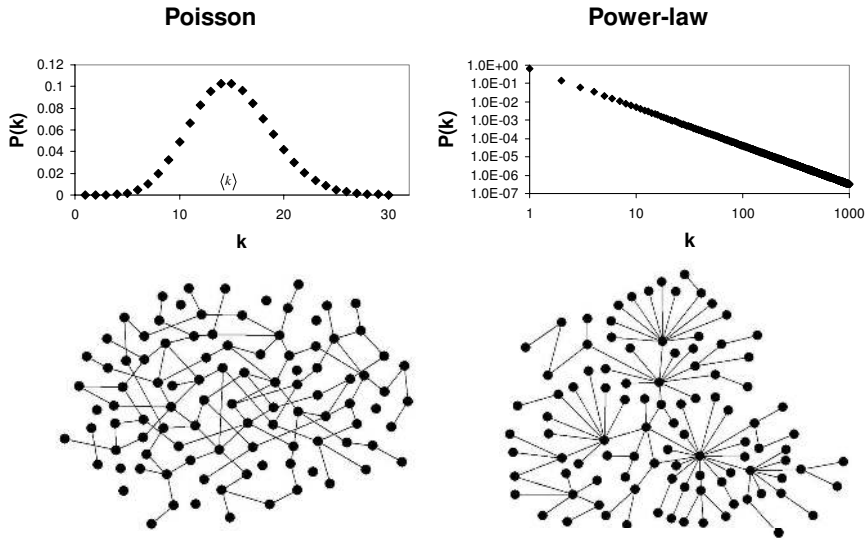


Figure 2.9. Comparison of networks with Poisson and power-law degree distribution of the same size. Note that the network with Poisson distribution is homogenous in node degree. Most of the nodes in the network have same degree which is close to the average degree of the network. However, the network with power-law degree distribution is highly heterogenous in node degree. There are few nodes with large degree and many nodes with a small degree

network k_i "stubs" sticking out of it and then pairs of these stubs are connected randomly to form complete edges [125]. Molloy and Reed [112] have proved that for a random graph with a degree distribution $p(k)$, a giant connected component emerges almost surely when $\sum_{k\geq 1} k(k-2)p(k) > 0$, provided that the maximum degree is less than $N^{1/4}$. Later, Aiello *et al.* [8, 9] introduced a two-parameter random graph model $P(\alpha, \gamma)$ for power-law graphs with exponent γ described as follows: Let n_k be the number of nodes with degree k, such that n_k and k satisfy $\log n_k = \alpha - \gamma \log k$. The total number of nodes in the network can be computed, noting that the maximum degree of a node in the network is $e^{\alpha/\gamma}$. Using the results from Molloy and Reed [112], they showed that there is almost surely a unique giant connected component if $\gamma < \gamma_0 = 3.47875....$ Whereas, there is no giant connected component almost surely when $\gamma > \gamma_0$.

Newman *et al.* [125] have developed a general approach to random graphs by using a generating function formalism [170]. The generating function for the degree distribution p_k is given by $G_0(x) = \sum_{k=0}^{\infty} p_k x^k$. This function captures all the information present in the original distribution since $p_k = \frac{1}{k!}\frac{\mathrm{d}^k G_0}{\mathrm{d}x^k}|_{x=0}$. The average degree of a randomly chosen node would be $< k >= \sum_k kp(k) = G_0^{'}(1)$. Further, this formulation helps in calculating other properties of the network [125]. For instance, we can approximately calculate the relation for the average path length of the network. Let us consider the degree of the node reached by following a randomly chosen edge. If the degree of this node is k then we are k times more likely to reach this node than a node of degree 1. Thus the degree distribution of the node arrived by a randomly chosen edge is given by kp_k and not p_k. In addition, the distribution of number of edges from this node (one less than the degree) q_k, is $\frac{(k+1)p_{k+1}}{\sum_k kp_k} = \frac{(k+1)p_{k+1}}{<k>}$. Thus, the generating function for q_k is given by $G_1(x) = \frac{\sum_{k=0}^{\infty}(k+1)p_{k+1}x^k}{k} = \frac{G_0^{'}(x)}{G_0^{'}(1)}$. Note that the distribution of the number of first neighbors of a randomly chosen node (degree of a node) is $G_0(x)$. Hence, the distribution of number of second neighbors from the same randomly chosen node would be $G_0(G_1(x)) = \sum_k p_k[G_1(x)]^k$. Here, the probability that any of the second neighbors is connected to first neighbors or to one another scales as N^{-1} and can be neglected in the limit of large N. This implies that the average number of second neighbors is given by $[\frac{\partial}{\partial x}G_0(G_1(x))]_{x=1} = G_0'(1)G_1'(1)$. Extending this method of calculating the average number of nearest neighbors, we find that the average number of m^{th} neighbors z_m, is $[G_1'(1)]^{m-1}G_0'(1) = [\frac{z_2}{z_1}]^{m-1}z_1$. Now, let us start from a node and find the number of first neighbors, second, third ... m^{th} neighbors. Assuming that all the nodes in the network can be reached within l steps, we have $1+\sum_{m=1}^{l} z_m = N$. As for most graphs $N \gg z_1$ and $z_2 \gg z_1$, we obtain the average path length of the network $l = \frac{N/z_1}{z_2/z_1} + 1$. The generating function formalism can further be extended to include other features such as directed graphs, bipartite graphs and degree correlations [122].

Another class of random graphs which are especially popular in modeling social networks is Exponential Random Graphs Models (ERGMs) or p* models [21, 67, 81, 155, 165]. The ERGM consists of a family of possible networks of N nodes in which each network G appears with probability $P(G) = \frac{1}{Z}\exp(-\sum_i \theta_i \epsilon_i)$, where the function Z is, $Z = \sum_G \exp(-\sum_i \theta_i \epsilon_i)$. This is similar to the Boltzmann

ensemble of statistical mechanics with Z as the partition function [122]. Here, $\{\epsilon_i\}$ is the set of observable's or measurable properties of the network such as number of nodes with certain degree, number of triangles etc. $\{\theta_i\}$ are adjustable set of parameters for the model. The ensemble average of a property ϵ_i is given as $\langle\epsilon_i\rangle = \sum_G \epsilon_i(G)P(G) = \frac{1}{Z}\epsilon_i \exp(-\sum_i \theta_i\epsilon_i) = \frac{\partial f}{\partial\theta_i}$. The major advantage of these models is that they can represent any kind of structural tendencies such as dyad and triangle formations. A detailed review of the parameter estimation techniques can be found in [21, 153]. Once the parameters $\{\theta_i\}$ are specified, the networks can be generated by using Gibbs or Metropolis-Hastings sampling methods [153].

2.2.2 Small-world networks

Watts and Strogatz [169] presented a small-world network model to explain the existence of high clustering and small average path length simultaneously in many real networks, especially, social networks. They argued that most of the real networks are neither completely regular nor completely random, but lie somewhere between these two extremes. The Watts-Strogatz model starts with a regular lattice on N nodes and each edge is rewired with certain probability p. The following is the algorithm for the model,

- Start with a regular ring lattice on N nodes where each node is connected to its first k neighbors.
- Randomly rewire each edge with a probability p such that one end remains the same and the other end is chosen uniformly at random. The other end is chosen without allowing multiple edges (more than one edge joining a pair of nodes) and loops (edges joining a node to itself).

The resulting network is a regular network when $p = 0$ and a random graph when $p = 1$, since all the edges are rewired (see figure 2.10). The above model is inspired from social networks where people are friends with their immediate neighbors such as neighbors on the street, colleagues at work etc (the connections in the regular lattice). Also, each person has a few friends who are a long way away (long-range connections attained by random rewiring). Later, Newman [118] proposed a similar model where instead of edge rewiring, new edges are introduced

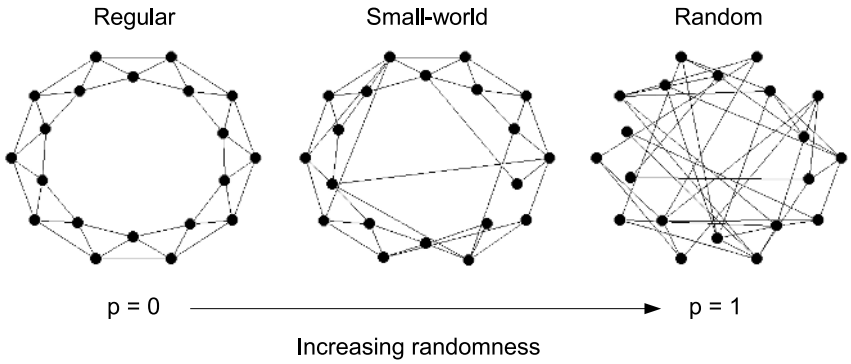


Figure 2.10. Illustration of the random rewiring process for the Watts-Strogatz model. This model interpolates between a regular ring lattice and a random network, without changing the number of vertices ($N = 20$) or edges ($E = 40$) in the graph. When $p = 0$ the graph is regular (each node has 4 edges), as p increases, the graph becomes increasingly disordered until $p = 1$, all the edges are rewired randomly. After Watts and Strogatz, 1998 [169].

with probability p. The clustering coefficient of the Watts-Strogatz model and the Newman model are

$$C_{WS} = \frac{3(k-1)}{2(2k-1)}(1-p)^3 \qquad C_N = \frac{3(k-1)}{2(2k-1)+4kp(p+2)}$$

respectively. This class of networks displays a high degree of clustering coefficient for small values of p since we start with a regular lattice. Also, for small values of p the average path length falls rapidly due to the few long-range connections. This co-existence of high clustering coefficient and small average path length is in excellent agreement with the characteristics of many real networks [118, 169]. The degree distribution of both models depends on the parameter p, evolving from a univalued peak corresponding to the initial degree k to a somewhat broader but still peaked distribution. Thus, small-world models are even more homogeneous than random graphs, which is not the case with real networks.

2.2.3 Scale-free networks

As mentioned earlier, many real networks including the World Wide Web [5, 15, 102], the Internet [64], peer-to-peer networks [145], metabolic networks [87], phone

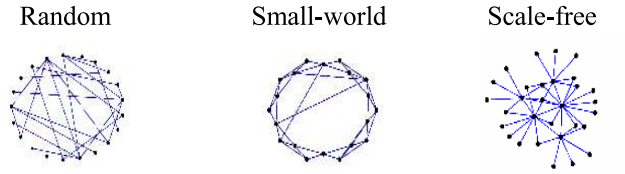
call networks [3, 8] and movie actor collaboration networks [13, 20, 26] are scale-free, that is, their degree distribution follows a power-law, $p(k) \sim k^{-\gamma}$. Barabási and Albert [26] addressed the origin of this power-law degree distribution in many real networks. They argued that a static random graph or Watts-Strogatz model fails to capture two important features of large-scale networks: their constant growth and the inherent selectivity in edge creation. Complex networks like the World-Wide Web, collaboration networks and even biological networks are growing continuously by the creation of new web pages, start of new researchers and by gene duplication and evolution. Moreover, unlike random networks where each node has the same probability of acquiring a new edge, new nodes entering the network do not connect uniformly to existing nodes, but attach preferentially to nodes of higher degree. This reasoning led them to define the following model,

- Growth: Start with small number of connected nodes say m_0 and assume that every time a node enters the system, m edges are pointing from it, where $m < m_0$.

- Preferential Attachment: Every time a new node enters the system, each edge of the newly entered node preferentially attaches to a already existing node i with degree k_i with the following probability,

$$\Pi_i = \frac{k_i}{\sum_j k_j}$$

It was shown that such a mechanism leads to a network with power-law degree distribution $p(k) = k^{-\gamma}$ with exponent $\gamma = 3$. These networks were called as scale-free networks because of the lack of a characteristic degree and the broad tail of the degree distribution. The average path length of this network scales as $\frac{\log(N)}{\log(\log(N))}$ and thus displays small world property. The clustering coefficient of a scale-free network is approximately $C \sim \frac{(\log N)^2}{N}$, which is a slower decay than $C =< k > N^{-1}$ decay observed in random graphs [37]. In the years following the proposal of the first scale-free model a large number of more refined models have been introduced, leading to a well-developed theory of evolving networks [14, 33, 56, 122]. The basic properties of most of these networks are similar to the three most prominent networks found in literature (see figure 2.11).



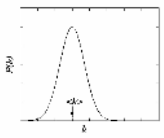
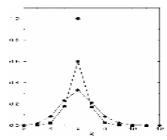
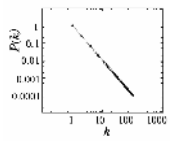
	Poisson	Peaked	Power Law
Degree Distribution			
Characteristic Path Length	Scales as $\log(N)$	Scales linearly with N for small *p*. And for higher *p* scales as $\log(N)$	Scales as $\log(N)/\log(\log(N))$
Clustering Coefficient	*p* (the connection probability)	High, but as $p \rightarrow 1$ behaves like a random graph	$((m-1)/2)*(\log(N)/N)$ where m is number of edges with which a node enters
Robustness to failures	Similar responses to both random and targeted attacks	Similar response as random networks. This is due to the fact that it has similar degree distribution as random networks	Highly resilient to random failures while being very sensitive to targeted attacks

Figure 2.11. Comparison of properties between random, small-world and scale-free networks

2.3 Dynamical processes in large-scale networks

The models discussed in section 2.2 are focused on explaining the evolution and growth process of many large real networks. They mainly concentrate on statistical properties of real networks and network modeling. *But the ultimate goal in studying and modeling the structure of complex networks is to understand and optimize the processes taking place on these networks.* For example, one would like to understand how the structure of the Internet affects its survivability against random failures or intentional attacks, how the structure of the WWW helps in efficient surfing or search on the web, how the structure of social networks affects the spread of viruses or diseases, etc. In other words, to design rules for optimization, one has to understand the interactions between the structure of the network and the processes taking place on the network. These principles will certainly help in redesigning or restructuring the existing networks and perhaps even help in designing a network from scratch. In the past few years, there has been tremendous

amount of effort by the research communities of different disciplines to understand the processes taking place on networks [14, 33, 56, 122]. In this section, we discuss a few dynamic processes and mainly concentrate on network resilience because of its high relevance to engineering systems. In the next section, we discuss the literature on decentralized search in networks which is the primary focus of this thesis.

2.3.1 Network resilience to node failures

All real networks are regularly subject to node/edge failures either due to normal malfunctions (random failures) or intentional attacks (targeted attacks) [12, 16]. Hence, it is extremely important for the network to be robust against such failures for proper functioning. Albert *et al.* [16] demonstrated that the topological structure of the network plays a major role in its response to node/edge removal. They showed that most of the real networks are extremely resilient to random failures. On the other hand, they are very sensitive to targeted attacks. They attribute this behavior to the fact that most of these networks are scale-free networks, which are highly heterogenous in node degree. Since a large fraction of nodes have small degree, random failures do not have any effect on the structure of the network. On the other hand, the removal of a few highly connected nodes that maintain the connectivity of the network, drastically changes the topology of the network. For example, consider the Internet: despite frequent router problems in the network, we rarely experience global effects. However, if a few critical nodes in the Internet are removed then it would lead to a devastating effect. Figure 2.12 shows the decrease in the size of the largest connected component for both scale-free networks and ER graphs, due to random failures and targeted attacks. ER graphs are homogenous in node degree, that is all the nodes in the network have approximately the same degree. Hence, they behave almost similarly for both random failures and targeted attacks (see figure 2.12(a)). In contrast, for scale-free networks, the size of the largest connected component decreases slowly for random failures and drastically for targeted attacks (see figure 2.12(b)).

Ideally, we would like to have a network which is as resilient as scale-free networks to random failures and as resilient as random graphs to targeted attacks.

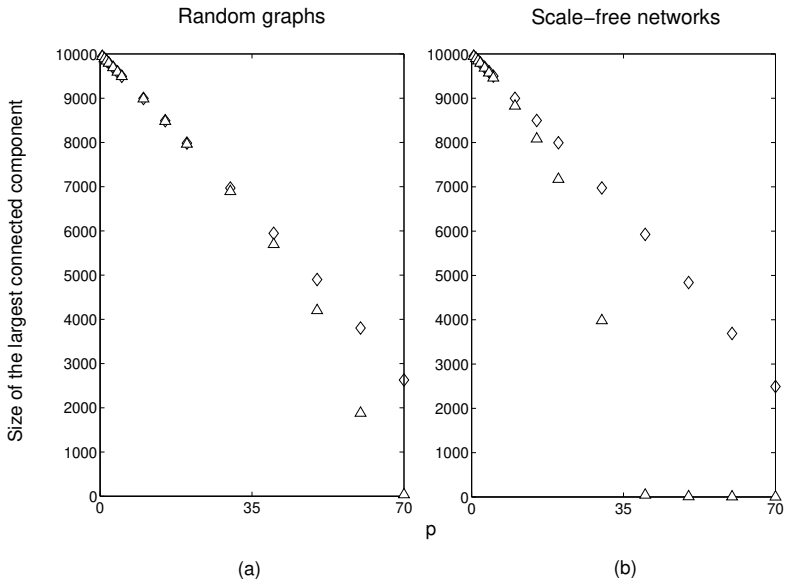


Figure 2.12. The size of the largest connected component as the percentage number of nodes (p) removed from the networks due to random failures ($\diamond$) and targeted attacks ($\triangle$). (a) ER graph with number of nodes (N) = 10,000 and mean degree $< k >= 4$; (b) Scale-free networks generated by Barabási-Albert model with N = 10,000 and $< k >= 4$. The behavior with respective to random failures and targeted attacks is similar for random graphs. Scale-free networks are highly sensitive to targeted attacks and robust to random failures.

To determine the feasibility of modeling such a network, Valente *et al.* [160] and Paul *et al.* [138] have studied the following optimization problem: "*What is the optimal degree distribution of a network of N nodes that maximizes the robustness of the network to both random failures and targeted attacks with the constraint that the number of edges remain the same?*"

Note that we can always improve the robustness by increasing the number of edges in the network (for instance, a completely connected network will be the most robust network for both random failures and targeted attacks). Hence the problem has a constraint on the number of edges. In [160], Valente *et al.* showed that the optimal network configuration is very different from both scale-free networks and random graphs. They showed that the optimal networks that maximize robustness for both random failures and targeted attacks have at most three distinct node degrees and hence the degree distribution is three-peaked. Similar results were

demonstrated by Paul *et al.* in [138]. Paul *et al.* showed that the optimal network design is one in which all the nodes in the network except one have the same degree, k_1 (which is close to the average degree), and one node has a very large degree, $k_2 \sim N^{2/3}$, where N is the number of nodes. However, these optimal networks may not be practically feasible because of the requirement that each node has a limited repertoire of degrees.

Many different evolutionary algorithms have also been proposed to design an optimal network configuration that is robust to both random failures and targeted attacks [49, 85, 150, 158, 161]. In particular, we [158] considered two other measures, *responsiveness* and *flexibility* along with robustness for random failures and targeted attacks, specifically for supply-chain networks. We defined responsiveness as the ability of network to provide timely services with effective navigation and measure it in terms of average path length of the network. The lower the average path length, the better is the responsiveness of the network. Flexibility is the ability of the network to have alternate paths for dynamic rerouting. Good clustering properties ensure the presence of alternate paths, and the flexibility of a network is measured in terms of the clustering coefficient. We designed a parameterized evolutionary algorithm for supply-chain networks and analyzed the performance with respect to these three measures. Through simulation we have shown that there exist trade-offs between these measures and proposed different ways to improve these properties. However, it is still unclear as to what would be the optimal configuration of such *survivable* networks. The research question would be "*what is the optimal configuration of a network of N nodes that maximizes the robustness to random failures, targeted attacks, flexibility, and responsiveness, with the constraint that the number of edges remain the same?*"

Until now, we have focussed on the effects of node removal on the static properties of a network. However, in many real networks, the removal of nodes will also have dynamic effects on the network as it leads to avalanches of breakdowns also called cascading failures. For instance, in a power transmission grid, the removal of nodes (power stations) changes the balance of flows and leads to a global redistribution of loads over all the network. In some cases, this may not be tolerated and might trigger a cascade of overload failures [94], as happened on August 10th 1996 in 11 US states and two Canadian provinces [148]. Models of cascades

of irreversible [116] or reversible [50] overload failures have demonstrated that removal of even a small fraction of highly loaded nodes can trigger global cascades if the load distribution of the nodes is heterogenous. Hence, cascade-based attacks can be much more destructive than any other strategies considered in [16, 83]. Later, in [115], Motter showed that a defence strategy based on a selective further removal of nodes and edges, right after the initial attack or failure, can drastically reduce the size of the cascade. Other studies on cascading failures include [41, 113, 114, 163, 166].

2.3.2 Detecting community structures

As mentioned earlier, community structures are typically found in many real networks. Finding these communities is extremely helpful in understanding the structure and function of the network. Sometimes the statistical properties of the community alone may be very different from the whole network and hence these may be critical in understanding the dynamics in the community. The following are some of the examples:

- *The World Wide Web*: Identification of communities in the web is helpful for implementation of search engines, content filtering, automatic classification, automatic realization of ontologies and focussed crawlers [19, 66].

- *Social networks*: Community structures are a typical feature of a social network. The behavior of an individual is highly influenced by the community he/she belongs. Communities often have their own norms, subcultures which are an important source of a person's identity [124, 164].

- *Biological networks*: Community structures are found in cellular [82, 146], metabolic [143] and genetic networks [171]. Identifying them helps in finding the functional modules which correspond to specific biological functions.

Algorithmically, the community detection problem is the same as the cluster analysis problem studied extensively by the OR community, computer scientists, statisticians, and mathematicians [78]. One of the major classes of algorithms for clustering is hierarchical algorithms which fall into two broad types, agglomerative and divisive. In an agglomerative method, an empty network (n nodes with no

edges) is considered and edges are added based on some similarity measure between nodes (for example, similarity based on the number of common neighbors) starting with the edge between the pairs with highest similarity. This procedure can be stopped at any step and the distinct components of the network are taken to be the communities. On the other hand, in divisive methods edges are removed from the network based on certain measure (for example, the edge with the highest betweenness centrality [124]). As this process continues the network disintegrates into different communities. Recently, many such algorithms are proposed and applied to complex networks [33, 51]. A comprehensive list of algorithms to identify community structures in complex networks can be found in [51] where Danon *et al.* have compared them in terms of sensitivity and computational cost.

Another interesting problem in community detection is to find a clique of maximum cardinality in the network. A clique is a complete subgraph in the network. In the network $G(V, E)$, let $G(S)$ denote the subgraph induced by a subset $S \subseteq V$. A network $G(V, E)$ is complete if each node in the network is connected to every other node, i.e. $\forall i, j \in V, \{i, j\} \in E$. A clique C is a subset of V such that the induced graph $G(C)$ is complete. The maximum clique problem has many practical applications in coding theory, computer vision, project selection, economics and integration of genome mapping data [40, 79, 131]. For instance, in [35], Boginski *et al.* solve this problem for finding the maximal independent set in the market graph which can form a base for forming a diversified portfolio. The maximum clique problem is known to be NP-hard [68] and details on various algorithms and heuristics can be found in [88, 131]. Further, if the network size is large, then the data may not fit completely inside the computer's internal memory. Then we need to use external memory algorithms and data structures [4] for solving the optimization problems in such networks. These algorithms use slower external memory (such as disks) and the resulting communication between internal memory and external memory can be a major performance bottleneck. In [3], using external memory algorithms, Abello *et al* proposed decomposition schemes that make large sparse graphs suitable for processing by graph optimization algorithms.

2.3.3 Spreading processes

Diffusion of an infectious disease, computer virus or information on a network constitute examples of spreading processes. In particular, the spread of infectious diseases in a population is called *epidemic spreading*. The study of epidemiological modeling has been an active research area for a long time and is heavily used in planning and implementing various prevention and control programs [53]. Recently, there has been a burst of activities on understanding the effects of the network properties on the rate and dynamics of disease propagation [14, 33, 56, 122]. Most of the earlier methods used the *homogenous mixing hypothesis* [22], which implies that the individuals who are in contact with susceptible individuals are uniformly distributed throughout the entire population. However, recent findings (section 2.1) such as heterogeneities in node degree, presence of high clustering coefficients, and community structures indicate that this assumption is far from reality. Later, many models have been proposed [14, 33, 46, 56, 122, 133, 136] which consider these properties of the network. In particular, many researchers have shown that incorporating these properties in the model radically changes the results previously established for random graphs. Other spreading processes which are of interest include spread of computer viruses [25, 108, 123], data dissemination on the Internet [92, 162], and strategies for marketing campaigns [104].

2.3.4 Congestion

Transport of packets or materials ranging from packet transfer in the Internet to the mass transfer in chemical reactions in cell is one of the fundamental processes occurring on many real networks. Due to limitations in resources (bandwidth), increase in number of packets (packet generation rate) may lead to overload at the node and unusually long delivery times, in other words, congestion in networks. Considering a basic model, Ohira and Sawatari [127] have shown that there exists a phase transition from a free flow to a congested phase as a function of the packet generation rate. This critical rate is commonly called "congestion threshold" and the higher the threshold, the better is the network performance with respect to congestion.

Many studies have shown that an important role is played by the topology and

routing algorithms in the congestion of networks [43, 52, 58, 59, 74, 75, 152, 154, 159]. Toroczkai *et al.* [159] have shown that on large networks on which flows are influenced by gradients of a scalar distributed on the nodes, scale-free topologies are less prone to congestion than random graphs. Routing algorithms also influence congestion at nodes. For example, in scale-free networks, if the packets are routed through the shortest paths then most of the packets pass through the hubs and hence causing higher loads on the hubs [69]. Singh and Gupte [152] discuss strategies to manipulate hub capacity and hub connections to relieve congestion in the network. Similarly many congestion-aware routing algorithms [43, 58, 59, 154] have been proposed to improve the performance. Sreenivasan et al. [154] introduced a novel static routing protocol which is superior to shortest path routing under intense packet generation rates. They propose a mechanism in which packets are routed through hub avoidance paths unless the hubs are required to establish the route. Sometimes when global information is not available, routing is done using local search algorithms. Congestion due to such local search algorithms and optimal network configurations are studied in [23].

2.4 Decentralized search in networks

One of the important research problems that has many applications in engineering systems is decentralized search in networks. Decentralized search is the process, in which a node tries to find a network path to a target node using only local information. Local information implies that each node has information only about its first, or perhaps second neighbors and it is not aware of nodes at a larger distance and how they are connected in the network. Let us suppose some required information such as computer files or sensor data is stored at the nodes of a distributed network or database. Then, in order to quickly determine the location of particular information, one should have efficient local (decentralized) search algorithms. Examples include routing of sensor data in wireless sensor networks [10, 142], locating data files in peer-to-peer networks [91, 175], and finding information in distributed databases [42].

Decentralized search in networks can be formulated in two types of networks. In the first type of network, the global position of a node cannot be quantified and it is

difficult to know whether a step in the search process is towards the target node or away from the target node. This makes the local search process even more difficult. One such kind of network is the peer-to-peer network, *Gnutella* [91], where the network structure is such that one may know very little information about the location of the target node. Here, when a user is searching for a file he/she does not know the global position of the node that has the file. Further, when the user sends a request to one of its neighbors, it is difficult to find out whether this step is towards the target node or away from it. Whereas in the second type of network, the global position of the target node can be quantified and each node has this information. This information will guide the search process in reaching the target node. For example, if we look at the network considered in Milgram's experiment each person has the geographical and professional information about the target node. All the intermediary people (or nodes) use this information as a guide for passing the messages. For lack of more suitable name, we call the networks of the first type as non-spatial networks and the second type as spatial networks.

2.4.1 Search in non-spatial networks

The traditional search methods in non-spatial networks are broadcasting or random walk. In broadcasting, each node sends the message to all its neighbors. The neighbors in turn broadcast the message to all their neighbors, and the process continues. Effectively, all the nodes in the network would have received the message at least once or maybe more. This could have devastating effects on the performance of the network. A hint on the potential damages of broadcasting can be viewed by looking at the Taylorsville NC, elementary school project [167]. Sixth-grade students and their teacher sent out a sweet email to all the people they knew. They requested the recipients to forward the email to everyone they know and notify the students by email so that they could plot their locations on a map. A few weeks later, the project had to be canceled because they had received about 450,000 responses from all over the world [167]. A good way to avoid such a huge exchange of messages is by doing a walk. In a walk, each node sends the message to one of its neighbors until it reaches the target node. The neighbor can be chosen in different ways depending on the algorithm. If the neighbor is chosen randomly with

equal probability then it is called *random search*, while in a *high degree search* the highest degree neighbor is chosen. Adamic *et al.* [6] have demonstrated that high degree search is more efficient than random search in networks with a power-law degree distribution (scale-free networks). High degree search sends the message to a more connected neighbor that has higher probability of reaching the target node and thus exploiting the presence of heterogeneity in node degree to perform better. They showed that the number of steps (s) required for the random search until the whole graph is revealed is $s \sim N^{3(1-2/\gamma)}$ and for the high-degree search it is $s \sim N^{(2-4/\gamma)}$. Clearly, for $\gamma > 2.0$, the number of steps taken by high-degree search scales with a smaller exponent than the random walk search. Since most real networks have power-law degree distribution with exponent (γ) between 2.1 and 3.0, high-degree search would be more effective in these networks.

However, these algorithms assume that the edges in the network are equivalent. But, the assumption of equal edge weights (which may represent the cost, bandwidth, distance, or power consumption associated with the process described by the edge) usually does not hold in real-world networks. Many researchers [17, 28, 29, 38, 70, 72, 76, 100, 120, 126, 137, 139, 174], have pointed out that it is incomplete to assume that all the edges are equivalent. These studies have shown that heterogeneity is prevalent in the capacity and strength of the interconnections and is critical in most real-world networks. For instance, sociologists have shown that the weak links that people have outside their close circle of friends play a key role in keeping the social system together [72, 120]. The Internet traffic [137] or the number of passengers in the airline network [24, 28, 76] are critical dynamical quantities that can be represented by using weighted edges. Similarly, the diversity of the predator-prey interactions and of metabolic reactions is considered as a crucial component of ecosystems and metabolic networks respectively [17, 100, 139]. Thus it is incomplete to represent real-world systems with equal interaction strengths between different pairs of nodes.

Further, the search strategies proposed for un-weighted networks may no longer be optimal in weighted networks. The total path length (p) in a weighted network for the path $1-2-3...-n$, is given by $p = \sum_{i=1}^{n} w_{i,i+1}$, where $w_{i.i+1}$ is the weight on the edge from node i to node $i+1$. Even though we can find a path with smaller number of hops, the total path length may be high if the weights on these edges

are different. Although, Goh et al [70] and Braunstein et. al. [38] have studied the optimal distance in weighted complex networks, they assumed that each node has the knowledge of the entire network. Braunstein et. al. considered two cases, namely, weak disorder and strong disorder for shortest paths. If the selection of a path is controlled by the sum of the edges (in case of costs) it corresponds to weak disorder. In some cases the edge with the maximum weight in the path can be the bottle-neck (in case of communication networks). If the selection of the path is controlled by minimizing the maximum weighted edge in the path, it corresponds to strong disorder. Braunstein has shown that in the case of strong disorder the optimal distance scales as $N^{1/3}$ in Erdos-Renyi and Watts-Strogatz networks. For scale-free networks he has shown that optimal distance (in strong disorder) scales as $N^{1/3}$ for $\tau > 4$ and as $N^{\frac{\tau-3}{\tau-1}}$ for $3 < \tau < 4$. Thus the small-world property is destroyed in the case of strong disorder if we introduce non-uniform weights to the edges. On the other hand, he has shown that in the case of weak disorder the small-world property is still preserved. Although it was shown that short paths exist, the question of weather the nodes can find these shorts paths using just the local information in weighted complex networks, is still unanswered. One of the problems we focus on is to obtain efficient algorithms for local search in weighted networks. In the next two chapters, we describe the research problem and give the details of the proposed methodology and results obtained.

2.4.2 Search in spatial networks

In spatial networks the nodes are embedded in a metric space and they are connected based on the metric distance. Here, the global position of the target node in the space can guide the search process to reach the target node more quickly. The problem of local search in spatial networks goes back to the famous experiment by Stanley Milgram [111] (discussed in section 2.1) illustrating the short distances in social networks. Another important observation of the experiment, which is even more surprising, is the ability of these nodes to find these short paths using just the local information. As pointed out by Kleinberg [95, 96, 97], this is not a trivial statement because most of the time, people have only local information in the network. That is the information about their immediate friends

or perhaps their friends' friends. They do not have the global information about the acquaintances of all people in the network. Even in Milgram's experiment, the people to whom he gave the letters have only local information about the entire social network. Still, from the results of the experiment, we can see that arbitrary pairs of strangers are able to find short chains of acquaintances between them by using only local information. Many models have been proposed to explain the existence of such short paths [14, 33, 56, 118, 122, 169]. However, these models are not sufficient to explain the second phenomenon. The observations from Milgram's experiment suggest that there is something more embedded in the underlying social network that guides the message implicitly from the source to the target. Such networks which are inherently easy to search are called *searchable networks*. Mathematically, a network is searchable if the length of the search path obtained scales logarithmically with the number of nodes N ($\sim logN$) or lesser. Kleinberg demonstrated that the emergence of such a phenomenon requires special topological features [95, 96, 97]. Considering a family of grid-based models that generalize the Watts-Strogatz [169] model, he showed that only one particular model among this infinite family can support efficient decentralized algorithms. He considered the network on a two dimensional lattice where two nodes (say u and v) are connected with a probability proportional to $[d(u,v)]^{-r}$. $d(u,v)$ is the number of lattice steps separating the nodes u and v. If u is at (i,j) and v is at (k,l) then $[d(u,v)] = |k-i| + |l-j|$. The geographical interpretation of this is that the persons who stay closer on the grid are more probable to know each other than those who live far apart. He showed that for a particular value of r i.e. when $r = 2$, a simple greedy search, where the node passes the message to the neighbor closest to the target node based on the grid distance, is able to give short paths, i.e. the delivery time is bounded by a polynomial in $\log N$. For any other values of r, the delivery time increases asymptotically in polynomial degree. He also extended this model to d-dimensional lattice, where greedy search find short paths if and only if $r = d$. He further extended this model to hierarchical networks [97], where, again, the network was proven to be searchable only for a specific parameter value. Unfortunately, the models given by Kleinberg represent only a very small subset of complex networks. Independently, Watts *et al.* presented another model based upon plausible hierarchical social structures [168], to explain the phenomena ob-

served in Milgram's experiment. The networks were shown to be searchable by a greedy search algorithm for a wide range of parameter space. Other works on decentralized search include [23, 98, 107, 109, 149, 151, 175]. Simsek and Jensen [151] use homophily between nodes and degree disparity in the network to design a better algorithm for finding the target node. However, finding an optimal way to combine location and degree information is yet to be investigated (see [98] for a review). Another interesting problem studied by Clauset and Moore [45], and by Sandberg [149], is the question of how real-world networks evolve to become searchable. They propose a simple feedback mechanism where the nodes continuously conduct decentralized searches, and in the process partially rewire the edges to form a searchable network.

In this thesis, we consider search in a family of parameterized spatial network models that are heterogenous in node degree. In this model, nodes are placed in an n-dimensional space and are connected, based on preferential attachment and geographical constraints, to form spatial scale-free networks. Preferential attachment to high degree nodes is believed to be responsible for the emergence of the power-law degree distribution observed in many real-world networks [26], and geographical constraints account for the fact that nodes tend to connect to nodes that are nearby. Many real-world networks such as the Internet [173] and the worldwide airline network [73], can be described by this family of spatial network models. Our objective is to design decentralized search algorithms for this type of network model and demonstrate that this simple model defines a class of searchable networks. In the next chapter, we describe the research problem in more detail.

CHAPTER

THREE

Problem description: Decentralized search in networks

In this thesis, we address the problem of decentralized search and routing in large-scale networks. We broadly divide the problem into

- Search in non-spatial networks
- Search in spatial networks

In the following pages, we describe the process of decentralized search and routing and present the details of the research problems.

Search and routing is one of the most important and prevalent processes in many real-world networks. Many a time one needs to transport raw material/computer files/messages from one node to another using the edges of the network. Further, it is critical that the paths used are optimal with respect to resources such as time and cost. Some examples include:

- *The Internet*: Millions of files/packets are routed everyday from one computer/server to another in the Internet. Due to limited availability of resources such as bandwidth, it is extremely important that they are routed using optimal paths.

- *Supply-chain network*: Supply chains involve complex webs of interactions and transportation of raw material/finished goods among suppliers, manufacturers, distributors, third-party logistics providers, retailers, and customers.

- *Social networks*: Search for a specific person using the social acquaintance network as in Milgram's experiment [111].

- *The WWW*: Search for certain information in the World Wide Web. One could search either by using search engines or by navigating one page to another using the hyperlinks on the current page.

- *Airline networks*: Traveling from one place to another using the U.S. airline network. One can obtain a choice of itineraries from the closest airport at the departure location to the closest airport at the destination location using various sources such as travel agents, airline offices or the World Wide Web.

- *Sensor networks*: Sending information from the sensor node to the sink node. It is critical to route the message along the path that consumes the least energy due to limited battery power in the sensors.

- *Peer-to-peer networks*: Search and routing of data files between the nodes in the peer-to-peer network.

- *Road networks*: Finding shortest path with respect to time taken or distance traveled from one place to another using the road network. Many services such as Google Maps or MapQuest provide the directions for the shortest path with respect to time or distance.

The problem of search and routing in networks can be approached in different ways depending upon the available information. If the global information of the network is available, i.e. how each and every node is connected in the network is known, one could use abundant number of algorithms available in literature for calculating the optimal paths [7, 48]. For example, one could use breadth first search (BFS) algorithm if all the edges in the network have equal edge weights or use Dijkstra's algorithm if the network has unequal non-negative edge weights.

3.1 Decentralized search

In some scenarios the node may not able to have access to the global information of the network as given below:

- *Social networks*: As observed in Milgram's experiment [111], people have information only about their immediate friends or perhaps their friends' friends. They do not have the global information about the acquaintances of all people in the network.

- *Decentralized networks*: Here, the nodes have limited access and can communicate only with the neighboring nodes. This may be either due to security or privacy reasons. One such example is the peer-to-peer network Gnutella [91], where the network structure is such that one may know very little information about the location of the target node. Here, when a user is searching for a file he/she does not know the global position of the node that has the file.

- *Dynamic (ad-hoc) and distributed networks*: The structure of the network is dynamic where nodes may go up/down and the weights on the edges change continuously. One such example is wireless sensor networks. Due to severe constraints on the battery power and bandwidth availability, it is not possible to update the configuration of the network at each node at regular intervals.

These particular type of large-scale distributed networks with limited information are becoming more prevalent due to advances made in different areas of engineering, especially in peer-to-peer networks and sensor networks technology. In such scenarios we need to have decentralized algorithms that can navigate through the network by using only local information. Here using local information would mean that a node has the information only about its neighbors and may be its neighbors' neighbors.

As discussed in chapter 2 this problem can be broadly classified to two types of networks. In the first type of network, the global position of a node cannot be quantified and it is difficult to know whether a step in the search process is towards the target node or away from the target node. Whereas in the second type of network, the global position of the target node can be quantified and

each node has this information. This information will guide the search process in reaching the target node. For lack of more suitable name, we call the networks of the first type as non-spatial networks and the second type as spatial networks.

3.2 Search in non-spatial networks

A simple algorithm in non-spatial networks is random search. In random search, a neighbor is chosen randomly with equal probability until it reaches the target node. Another algorithm is high-degree search, where a neighbor with the highest degree is chosen. Adamic *et al.* [6] have demonstrated that high degree search is more efficient than random search in networks with a power-law degree distribution (scale-free networks). High degree search sends the message to a more connected neighbor that has higher probability of reaching the target node and thus exploiting the presence of heterogeneity in node degree to perform better. However, they assume that the edges in the network are equivalent which does not hold in many real-world networks [17, 28, 29, 38, 70, 72, 76, 100, 120, 126, 137, 139, 174]. In this thesis, we address the following problem of search in weighted complex networks.

3.2.1 Non-spatial network

Consider a network $G(N, E)$ on N nodes with a set of E edges and has the following properties:

1. Its node degree distribution follows a power-law in with exponent varying from 2.0 to 3.0. Although we discuss the search algorithms for networks with Poisson degree distribution (ER random graphs), we concentrate more on power-law networks since most of the real world networks are found to exhibit this behavior [14, 33, 56, 122].

2. It has non-uniformly distributed weights on the edges. Here the weights signify the cost/time taken to pass the message/query. Hence, smaller weights correspond to shorter/better paths. We consider different distributions like Beta, uniform, exponential and power-law.

3. It is unstructured and decentralized. That is, each node has information only about its first and second neighbors and no global information about the target is available. Also, the nodes can communicate only with their immediate neighbors.

4. The topology is dynamic (ad-hoc) while still maintaining its statistical properties.

Here, each node has information about the first and second neighbors and the weights of the edges connecting them. The position of the target node is unknown. The research problem is to design efficient search algorithms using this information in the above mentioned networks.

3.3 Search in spatial networks

We consider search in a family of parameterized spatial network models that are heterogenous in node degree. In this model, nodes are placed in an n-dimensional space and are connected, based on preferential attachment and geographical constraints, to form spatial scale-free networks (see figure 3.1). Preferential attachment to high degree nodes is believed to be responsible for the emergence of the power-law degree distribution observed in many real-world networks [26], and geographical constraints account for the fact that nodes tend to connect to nodes that are nearby. Many real-world networks such as the Internet [173] and the worldwide airline network [73], can be described by this family of spatial network models.

Our objective is to design decentralized search algorithms for this type of network model and demonstrate that this simple model defines a class of searchable networks. Each node has information about the position of the target node, the position of its neighbors, and the degree of its neighbors. Using this information, the start node, and consecutively each node receiving the message, passes the message to one of its neighbors based on the search algorithm until it reaches the target node. The following section gives the details of the network model.

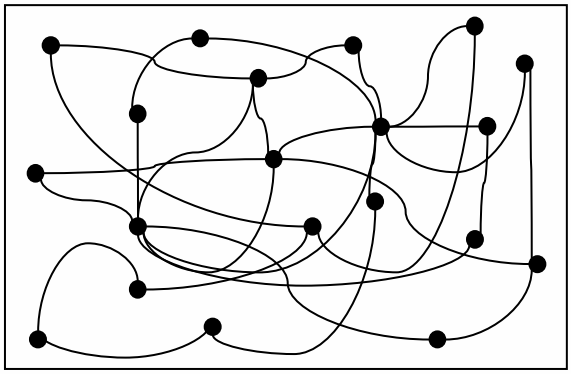


Figure 3.1. Illustration of a spatial network in two-dimensional space. Nodes are placed in an 2-dimensional space and are connected, based on preferential attachment and geographical constraints, to form spatial scale-free network

3.3.1 Spatial network model

The spatial network model we consider incorporates both preferential attachment and geographical constraints. At each step during the evolution of the spatial network model one of the following occurs [55]:

- with probability p, a new edge is created between two existing nodes in the network;
- with probability $1-p$, a new node is added and connected to m existing nodes in the network, with the constraint that multiple edges are not formed.

In both cases, the degrees of the nodes and the distances between them are considered when forming a new edge. In the first case, two nodes i and j are selected according to

$$\Pi_{ij} \propto \frac{k_i k_j}{F(d_{ij})},$$

where k_i is the degree of node i, d_{ij} is the Euclidian distance between node i and node j and $F(d_{ij})$ is an increasing function of d_{ij}. A new node i is uniformly and randomly placed in an n-dimensional space and is connected to a pre-existing node j with probability

$$\Pi_j \propto \frac{k_j}{F(d_{ij})}.$$

The above process is simulated until the number of nodes in the network is N. Let the network generated be $G(N, p, m, F, n)$. Here, the preferential attachment mechanism leads to a power-law degree distribution where the exponent can be tuned by changing the value of p [55] (see figure 5.2(a)). $F(d)$ controls the truncation of the power-law decay, and if $F(d)$ increases rapidly, then the power-law decay regime can disappear altogether [31]. Two widely-used functions for $F(d)$ are d^r [173] and $\exp(d/d_{char})$ [31].

3.4 Sensor networks and other applications

In this section, we discuss some of the applications where the above two problems are applicable. Recent advances in technology have enabled the development of low-cost, low-power sensor nodes that are small in size and can communicate only with in short distances [10]. These tiny sensors, which are capable of sensing, communicating and data processing has led to the idea of sensor networks. Sensor networks consist of large number of tiny sensors that coordinate amongst themselves to achieve a larger sensing task. Hundreds to several thousands of nodes are deployed within a small area, which is called sensor field (see figure 3.2).

The main task of a sensor node in a sensor field is to detect events, perform local data processing and then transmit the data. Due to severe hardware constraints (battery power and scope of transmission), these sensors can communicate only with sensors within their vicinity. Also, the nodes can switch between active and inactive states at random times which is a main source of uncontrolled dynamics in the topology of the network. This is due to exhaustion of the power and being charged by renewable sources. Several aspects of these sensor networks lead to many design challenges, which are different from the ones posed by the conventional networks. Further, the sheer number of these tiny sensors would present unique challenges in the design of unattended autonomous sensor networks [63]. These challenges demand design and operation techniques which are different from the ones used for conventional network applications. One of the challenges is the need for decentralized routing algorithms which are extremely energy-efficient.

The sensor nodes are usually scattered in the sensor field. Each of these sensor nodes has to route the information collected by it to the sink node and then to

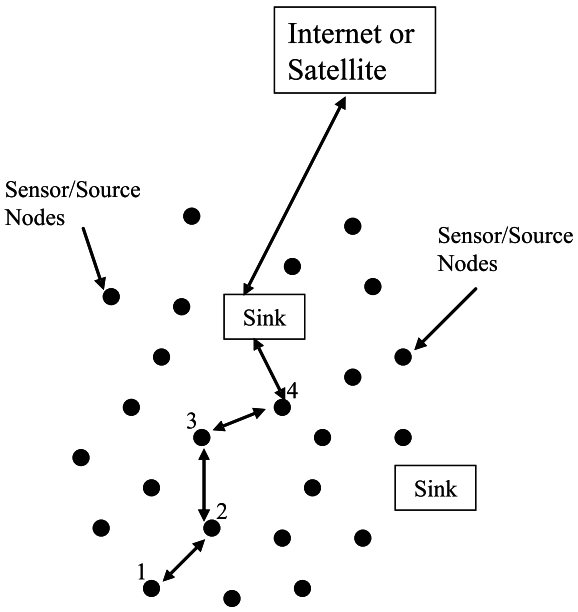


Figure 3.2. Sensor field

the end users (see figure 3.2). The sink may route back to the end user using the Internet or a satellite. The conventional way to route or search in the above mentioned scenario is to broadcast the information to all the neighbors (as shown in the figure 3.3) or send the message to a randomly selected neighbor [101]. But we clearly see that this is not an efficient way, especially when the nodes operate under severe power constraints. One of the major drawback of *classical flooding or broadcasting* is implosion. During flooding, a node always sends the message to all the neighbors irrespective of whether or not the neighbor has already received the message which will lead to many redundant transmission. For example in network shown in figure 3.4, node 1 starts broadcasting the message to all its neighbors. In the next step, node 2 and node 3 sends the message to all its neighbors except to node 1. Hence, node 4 receives two copies of the same data. Also, whenever data reaches a high degree node, then more copies of the same data are transmitted in the network.

Many decentralized routing algorithms have been proposed in the literature which perform better than flooding [10, 11]. However, none of these algorithms try to exploit the structural properties of the network. Most of the algorithms in wireless sensor networks literature find a path to the target node either by

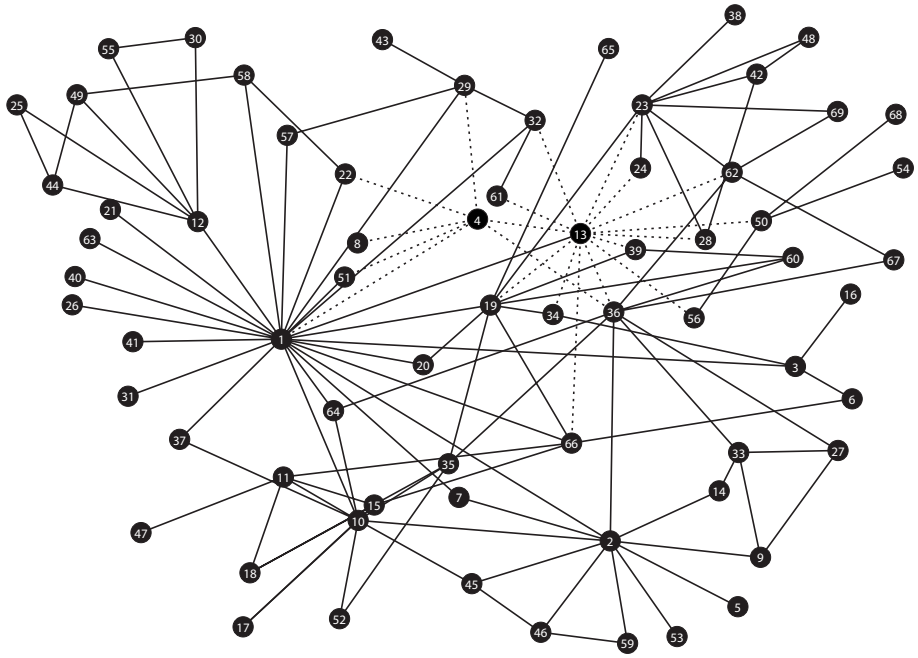


Figure 3.3. Broadcasting in a complex network. Node 4 broadcasts the message to all the neighbors and then all the neighbors do the same. The broadcasted edges are shown as dotted lines. In figure we show only one of the node 4's neighbor 13 broadcasting for clarity. We clearly see that this search algorithm is inefficient.

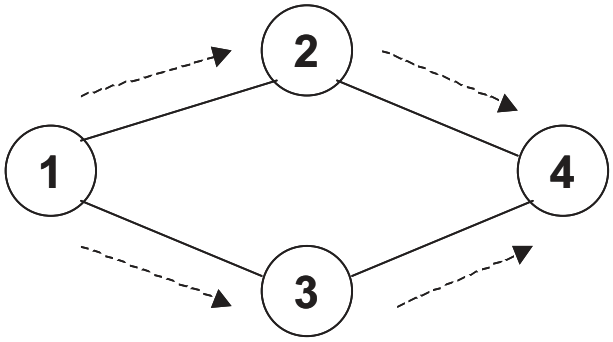


Figure 3.4. Implosion in classical flooding. In this network, node 1 starts broadcasting the message to all its neighbors. Then, in classical flooding node 2 and node 3 sends the message to all its neighbors except to node 1. Hence, two copies of the same data have been received to node 4.

broadcasting or random walk and then concentrate on efficient routing of the data from start node to the end node [10, 86]. As we will see in next two chapters, the properties of the networks have significant effect on the search process. Hence, the algorithms in wireless sensor networks literature could be integrated with the results obtained from the above two problems for better performance.

Another important application of decentralized search is peer-to-peer networks such as Gnutella. Gnutella [91] is a decentralized and unstructured peer-to-peer network used for sharing information between different users. It does not have any centralized server which will index all the users (represented as nodes) and files available. Each node is connected to few other nodes and has information about the files available with the neighbors. If the files are not available with the neighbors, they are searched by sending a query to the neighboring nodes. In this network, a user (represented as a node) can join any part of the network in real time and leave the network at anytime. Hence, Gnutella is an instance of a large-scale network with uncontrolled dynamics. When a new node joins the network it brings some amount of network capacity (in terms of files shared) with it. When it leaves the network, the nodes that are connected to the departing node clean up the memories to remove the information about the departing node. Currently, the search queries are broadcasted to all the neighbors, which is inefficient with respect to the amount of bandwidth consumed. The results from problem 1 would help in optimal usage of available bandwidth and to decrease congestion in the network. Similarly, decentralized search algorithms are required in many other networks where information is distributed and availability is constrained.

In the next chapter, we discuss the methodology for search in non-spatial networks and present the findings. In chapter 5, we study the decentralized search for spatial networks and summarize the findings.

CHAPTER

FOUR

Search in non-spatial networks

We simulated and analyzed many search algorithms to study the complex tradeoffs presented by efficient local search in weighted complex networks. Among the search algorithms employed is a novel algorithm based on the local betweenness centrality (LBC) of nodes. Betweenness centrality (also called load), first developed in the context of social networks [164], has been recently adapted to optimal transport in weighted complex networks by Goh *et al.* [70]. To determine a node's betweenness as defined by Goh *et al.*, one would need to have the knowledge of the entire network. Here we define a local-parameter called local betweenness centrality (LBC) which only uses information on the first and second neighbors of a node, is discussed in detail in the next section. Later, we define a search algorithm based on this local parameter and show that it utilizes both the heterogeneity in node degree and edge weights to perform the best in large class of networks.

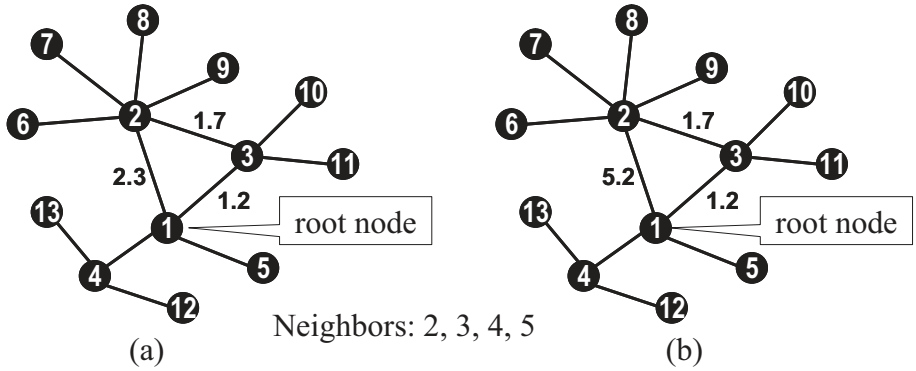
4.1 Local betweenness centrality

We assume that each node in the network has information about its first and second neighbors. For calculating the LBC of the neighbors of a given node we consider the local network formed by that node (which we will call the root node), its first and second neighbors. Then, the betweenness centrality, defined as the fraction of shortest paths going through a node [122], is calculated for the first neighbors in

this local network. Let $L(i)$ be the LBC of a neighbor node i in the local network. Then $L(i)$ is given by

$$L(i) = \sum_{\substack{s \neq n \neq t \\ s,t \in \text{ local network}}} \frac{\sigma_{st}(i)}{\sigma_{st}},$$

where σ_{st} is the total number of shortest paths (the path over which the sum of weights is minimum) from node s to t. $\sigma_{st}(i)$ is the number of these shortest paths passing through i. If the LBC of a node is high, it implies that this node is central in the local network. Intuitively, we can see that the LBC of a neighbor depends on both its degree and the weight of the edge connecting it to the root node. For example, let us consider the networks in figure 4.1(a) and figure 4.1(b). Suppose that these are the local networks of node 1. In the network in Figure 1(a), node 2 has the highest degree among the neighbors of node 1 (i.e. nodes 2, 3, 4 and 5). All the shortest paths from the neighbors of node 2 (6, 7, 8 and 9) to other nodes have to pass through node 2. Hence, we see that higher degree for a node definitely helps in obtaining a higher LBC.



(a) LBC: 2 → **76.0**, 3 → 42.0, 4 → 42.0, 5 → 0.0

(b) LBC: 2 → 76.0, 3 → **92.0**, 4 → 42.0, 5 → 0.0

Figure 4.1. (a) In this configuration, neighbor node 2 has a higher LBC than other neighbors 3, 4 and 5. This depicts why higher degree for a node helps in obtaining higher LBC. (b) However, in this configuration the LBC of the neighbor node 3 is higher than neighbors 2, 4 and 5. This is due to the fact that the edge connecting 1 and 2 has a larger weight. These two configurations show that the LBC of a neighbor depends both on the edge weight and the node degree. In both cases, edge-weights other than those shown in the figure are 1.

Now consider a similar local network but with a higher weight on the edge from

2 to 1 as shown in figure 4.1(b). In this network all the shortest paths through node 2 will also pass through node 3 (2-3-1) instead of going directly from node 2 to node 1. Hence, the LBC of the neighbor node 3 will be higher than neighbor 2. Thus we clearly see that the LBCs of the neighbors of node 1 depend on both the neighbors' degrees and the weights on the edges connecting them. Note that a neighbor having the highest degree or the smallest weight on the edge connecting it to root node does not necessarily imply that it will have the highest LBC.

4.2 Different search algorithms

In un-weighted scale-free networks, Adamic et. al. [6] have shown that high degree search is efficient. Thus one expects that in weighted power-law networks, an efficient search algorithm should consider both the edge weights and node degree. We investigated the following set of search algorithms given in the order of the amount of information required..

1. *Choose a neighbor randomly*: Here, a node tries to reach the target by passing the message/query to a randomly selected neighbor.

2. *Choose the neighbor with smallest edge weight*: Here, a node passes the message along the edge with minimum weight. The idea behind this algorithm is that by choosing a neighbor with minimum edge weight the expected distance traveled would be less.

3. *Choose the best-connected neighbor*: Here a node passes the message to the neighbor which has the highest degree. The idea here is that by choosing a neighbor which is well-connected, there is a higher probability of reaching the target node. Note that this algorithm takes the least number of hops to reach the target [6].

4. *Choose the neighbor with the smallest average weight*: Here a node passes the message to the neighbor which has the smallest average weight. The average weight of a node is the average weight of the edges incident on that node. The idea here is similar to the second algorithm. Instead of passing

the message greedily along the least weighted edge, the algorithm passes to the node that has the minimum average weight.

5. *Choose the neighbor with the highest LBC*: Here a node passes the message to the neighbor which has the highest LBC. A neighbor with highest LBC would imply that many shortest paths in the local network pass through this neighbor and the node is central in the local network. Thus, by passing the message to this neighbor, the probability of reaching the target node quicker is higher.

Note that the algorithm which depends on LBC utilizes slightly more information than algorithm 4, namely the edge weights between second neighbors, but it is considerably more informative, it reflects the heterogeneities in both edge weights and node degree. Thus we expect that this search will perform better than the others, that is, it will give smaller path lengths than the others.

4.3 Simulation and Analysis

Simulations on a random network with a Poisson and power-law degree distributions were used for comparing the search algorithms. For homogeneous networks we used Poisson random network model given by Erdos-Renyi [60]. We considered a network on N nodes where two nodes are connected with a connection probability p. For power-law networks, we considered different values of degree exponent τ ranging from 2.0 to 3.0 and a degree range of $2 < k < m \sim N^{1/\tau}$ and generated the network using the method given by Newman [117]. Once the network was generated, we extracted the largest connected component, shown to always exist for $2 < \tau < 3.48$ [8] and in ER networks for $p > \frac{1}{N}$ [36]. We did our analysis on this largest connected component that contains the majority of the nodes after verifying that the degree distribution of this largest connected component is nearly the same as in the original graph. The weights on the edges were generated from different distributions like Beta, uniform, exponential and power-law. We considered these distributions in the increasing order of their variances to understand how the heterogeneity in edge weights affects different search algorithms.

Further, we randomly chose K pairs (source and target) of nodes. The source, and consecutively each node receiving the message, sends the message to one of its neighbor depending on the search algorithm. The search continues until the message reaches the node whose neighbor is the target node. In order to avoid passing the message to a neighbor that has already received it, a list l_i of all the neighbors that received the message is maintained at each node i. During the search process, if node i passes the message to its neighbor j, which does not have any more neighbors that are not in the list l_j, then the message is routed back to the node i. This particular neighbor j is marked to note that this node cannot pass the message any further. The average path distance was calculated for each algorithm from the paths obtained for these K pairs. We repeated this simulation for 10 to 50 instances of the Poisson and power-law networks depending on the size of the network.

For the initial step, we used ER random graphs to compare different search algorithms. The weights on the edges were generated from an exponential distribution with mean 5 and variance 25. Table 4.1 compares the performance of each algorithm for the networks of size 500, 1000, 1500 and 2000 nodes. We took the connection probability to be $p = .004$ and hence a giant connected component always exists [36]. From Table 4.1, it is evident that the algorithm which passes the message to the neighbor with the least edge weight is better than all the other algorithms in homogeneous networks. Remarkably, an algorithm that needs less information than other algorithms (3, 4 and 5), performed best, while high degree search and LBC did not perform well since the network is highly homogenous in node degree.

However, if we decrease the heterogeneity in edge weights (use a distribution with lesser variance), we observe that high LBC search performs best (see the column with Beta distribution in the Table 4.2). In conclusion, when the heterogeneity of edge weights is high compared to the relative homogeneity of node degrees, the search algorithms which are purely based on edge weights would perform better. However, as the heterogeneity of the edge weights decrease the importance of edge weights decreases and algorithms which consider both edge weights and node degree perform better.

Next we investigated the algorithms on power-law networks. Figure 4.2 shows

Table 4.1. Comparison of search algorithms in a Poisson random network. The edge weights were generated randomly from an exponential distribution with mean 5 and variance 25. The values in the table are the average path distances obtained for each search algorithm in these networks. The algorithm which passes the message to the neighbor with the least edge weight performs the best.

Search algorithm	500 nodes	1000 nodes	1500 nodes	2000 nodes
Random walk	1256.3	2507.4	3814.9	5069.5
Minimum edge weight	**597.6**	**1155.7**	**1815.5**	**2411.2**
Highest degree	979.7	1923.0	2989.2	3996.2
Minimum average node weight	832.1	1652.7	2540.5	3368.6
Highest LBC	864.7	1800.7	2825.3	3820.9

Table 4.2. Comparison of search algorithms in a Poisson random network with 2000 nodes. The table gives results for different edge weight distributions. The mean for all the distributions is 5 and variance is σ^2. The values in the table are the average path lengths obtained for each search algorithm in these networks. When the weight heterogeneity is high, the minimum edge weight search algorithm was the best. However, when the heterogeneity of edge weights is low, then LBC performs better.

Search algorithm	Beta $\sigma^2 = 2.3$	Uniform $\sigma^2 = 8.3$	Exp. $\sigma^2 = 25$	Power-law $\sigma^2 = 4653.8$
Random walk	1271.91	1284.9	1253.68	1479.32
Minimum edge weight	1017.74	767.405	577.83	562.39
Highest degree	994.64	1014.05	961.5	1182.18
Minimum average node weight	1124.48	954.295	826.325	732.93
Highest LBC	980.65	968.775	900.365	908.48

the scaling of different algorithms for power-law networks with exponent 2.1.

As conjectured, the search algorithm that uses the information about both the edge weights and nodes' degrees (the high LBC search) performed better than the others. A similar phenomenon was observed for different exponents of the power-law network (see Table 4.3). Except for the power-law exponent 2.9, the high LBC search was consistently better than others.

We observe that as the heterogeneity in the node degree decreases (i.e. as power-law exponent increases), the difference between the high LBC search and other algorithms decreases. When the exponent is 2.9, the performance of LBC, minimum edge weight and high degree searches were almost the same. Note that

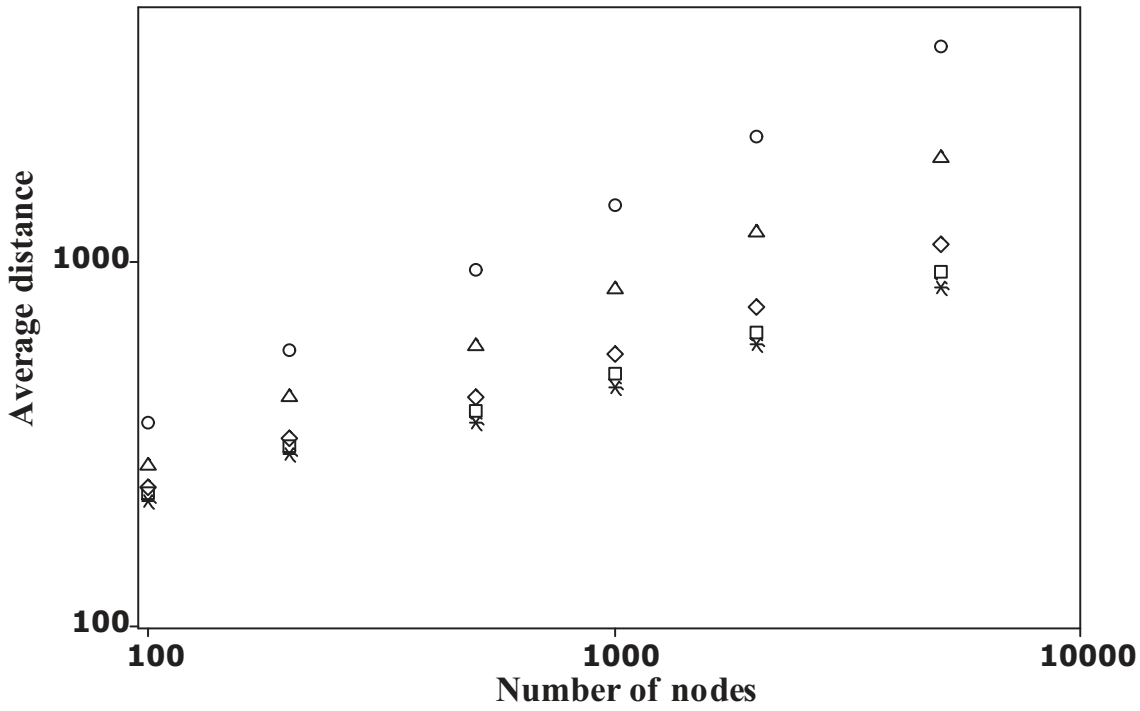


Figure 4.2. Scaling for search algorithms in power-law networks with exponent 2.1. The edge weights are generated from an exponential distribution with mean 10 and variance 100. The symbols represent random walk (○) and search algorithms based on minimum edge weight (□), high degree (◇), minimum average node weight (△) and high LBC (∗).

Table 4.3. Comparison of search algorithms in power-law network on 2000 nodes with different power-law exponents. The edge weights are generated from an exponential distribution with mean 5 and variance 25. The values in the table are the average path lengths obtained for each search algorithm in these networks. LBC search, which reflects both the heterogeneities in edge weights and node degree, performed the best for all power-law exponents. The systematic increase in all path lengths with the increase of the power-law exponent τ is due to the fact that the average degree of the network decreases with τ.

power-law exponent = Search algorithm	2.1	2.3	2.5	2.7	2.9
Random walk	1108.70	1760.58	2713.11	3894.91	4769.75
Minimum edge weight	318.95	745.41	1539.23	2732.01	3789.56
Highest degree	375.83	761.45	1519.74	2693.62	3739.61
Minimum average node weight	605.41	1065.34	1870.43	3042.27	3936.03
Highest LBC	**298.06**	**707.25**	**1490.48**	**2667.74**	**3751.53**

when the network becomes homogeneous in node degree the minimum edge weight search performed better than high LBC search (Table 4.1). This implies that similar to high degree search [6], the effectiveness of high LBC search also depends on the heterogeneity in node degree.

Table 4.4 shows the performance of all the algorithms on a power-law network (exponent 2.1) with different edge weight distributions. The percentage values in the brackets show by how much the average distance for that search is higher than the average distance obtained by the high LBC search. As in random graphs, we observe that the impact of edge weights on search algorithms increases as the heterogeneity of the edge weights increase. For instance, when the variance (heterogeneity) of edge weights is small, high degree search is better than the minimum edge weight search. On the other hand, when the variance (heterogeneity) of edge-weights is high, the minimum edge weight algorithm is better than high degree search. A. In each case, the high LBC search which considers both edge weights and node degree always out-performed the other algorithms. Thus, it is clear that in power-law networks, irrespective of the edge weight distribution and the power-law exponent, high LBC search always performs better than the other algorithms (Tables 4.3 and 4.4). Also, note that the minimum average node weight algorithm uses only slightly less information than LBC search. However, LBC search consistently and significantly outperforms it (see tables 4.1, 4.2, 4.3, and 4.4). This implies that LBC search uses the information correctly.

4.3.1 Comparison of high degree search and high LBC search

Figure 4.3 gives a pictorial comparison of the behavior of high degree and high LBC search as the heterogeneity of the edge weights increase (based on the results shown in Table 4.4). Further, figure 4.4 plots the scaling of the high degree and high LBC algorithms with network size for different heterogeneities in edge-weights. We notice that the scaling of high-degree search is same irrespective of the edge-weight distribution. Whereas, the scaling of LBC search decreases as the the heterogeneity in edge-weight increases. This implies that when the heterogeneity in edge-weights is high, the LBC search utilizes low weight edges for navigation. Since many studies [17, 18, 28, 29, 30, 38, 57, 65, 70, 72, 76, 89, 100, 105, 106, 120,

Table 4.4. Comparison of different search algorithms in power-law networks with exponent 2.1 and 2000 nodes with different edge weight distributions. The mean for all the edge weight distributions is 5 and the variance is σ^2. The values in the table are the average distances obtained for each search algorithm in these networks. The values in the brackets show the relative difference between average distance for each algorithm with respect to the average distance obtained by the LBC algorithm. LBC search, which reflects both the heterogeneities in edge weights and node degree, performed the best for all edge weight distributions.

Search algorithm	Beta $\sigma^2 = 2.3$	Uniform $\sigma^2 = 8.3$	Exp. $\sigma^2 = 25$	Power-law $\sigma^2 = 4653.8$
Random walk	1107.71 (202%)	1097.72 (241%)	1108.70 (272%)	1011.21 (344%)
Minimum edge weight	704.47 (92%)	414.71 (29%)	318.95 (7%)	358.54 (44%)
Highest degree	379.98 (4%)	368.43 (14%)	375.83 (26%)	466.18 (59%)
Minimum average node weight	1228.68 (235%)	788.15 (145%)	605.41 (103%)	466.18 (88%)
Highest LBC	**366.26**	**322.30**	**298.06**	**247.77**

126, 129, 137, 139, 174], have shown that there exists large heterogeneity in the capacity and strengths of the interconnections in the real networks, it is important that local search is based on LBC rather than high degree as shown by Adamic et. al. [6].

Note that LBC has been adopted from the definition of betweenness centrality (BC) which requires the global knowledge of the network. BC is defined as the fraction of shortest paths among all nodes in the network that pass through a given node and measures how central the node is for optimal transport in complex networks In un-weighted scale-free networks there exists a scaling relation between node betweenness centrality and degree, $BC\ k^{\eta}$[69]. This implies that the higher the degree, the higher is the BC of the node. This may be the reason why high degree search is optimal in un-weighted scale-free networks (as shown by Adamic et al [6]). However, Goh et. al. [70] have shown that no scaling relation exists between node degree and betweenness centrality in weighted complex networks. It will be interesting to assess the relationship between local and global betweenness centrality in our future work.

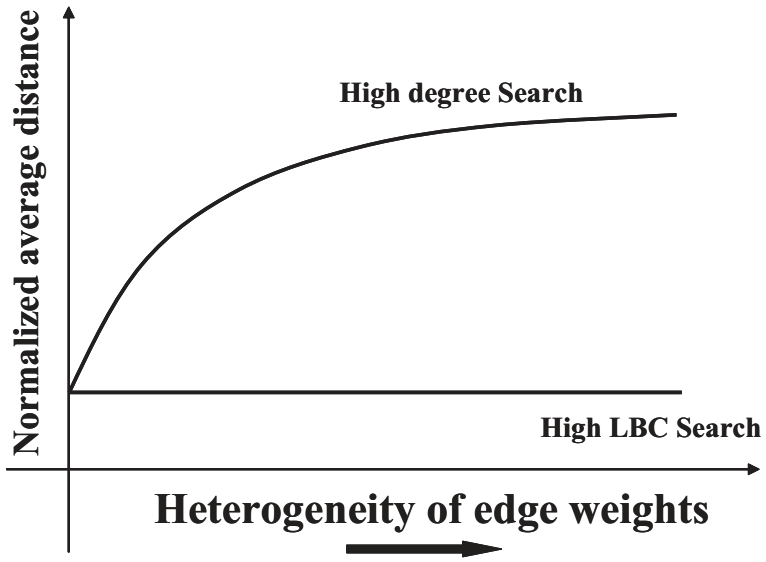


Figure 4.3. The pictorial comparison of the behavior of high degree and high LBC search as the heterogeneity of edge weights increases in power-law networks. Note that average distances are normalized with respect to high LBC search.

4.3.2 LBC on un-weighted networks

In this section, we demonstrate that the neighbor with the highest LBC is same as the neighbor with the highest degree in un-weighted networks. Hence, high LBC search would be same as high degree search in un-weighted networks. In un-weighted networks, there is a scaling relation between the BC of a node and its degree, as $BC \sim k^{\eta}$ [69]. However, this does not imply that in an un-weighted local network the neighbor with highest LBC is the same as the neighbor with the highest degree. Here we show that in most cases the highest degree and the highest LBC neighbors coincide. First, let us consider a tree-like local network without any loops similar to the network configuration shown in figure 4.5(a).

In a local network, there are three types of nodes, namely, root node, first neighbors and second neighbors. Let the degree of the root node be d and the degree of the neighbors be $k_1, k_2, k_3 ... k_d$. The number of nodes (n) in the local network is $n = 1 + \sum_{j=1}^{d} k_j$ (one root node, d first neighbors and $\sum_{j=1}^{d}(k_j - 1)$ second neighbors). In a tree network there is a single shortest path between any pair of nodes s and t, thus $\sigma_{st}(i)$ is either zero or one. Then the LBC of a first

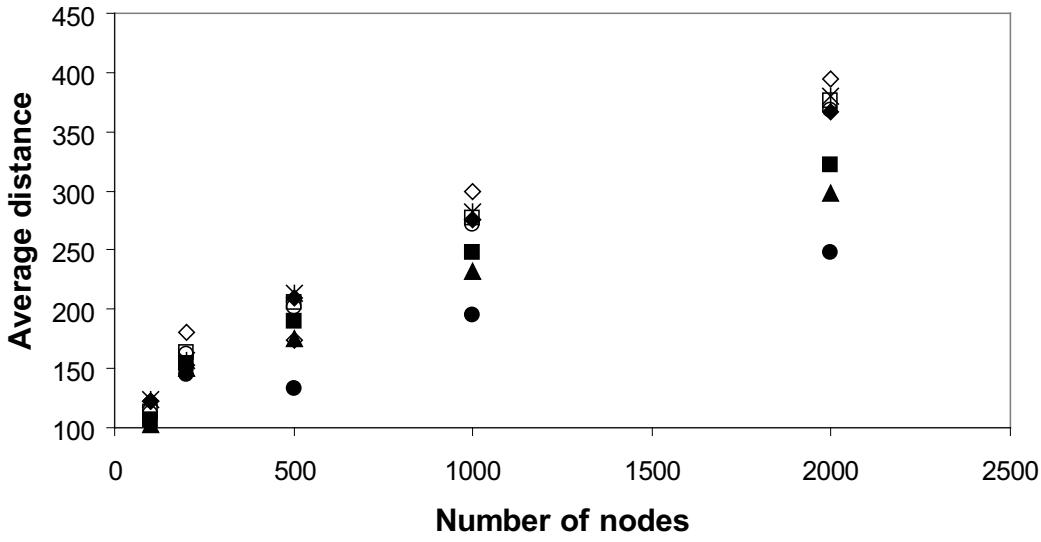


Figure 4.4. Scaling of high degree and high LBC search with network size for different heterogeneities in edge weights. The filled symbols represent LBC search for different edge weight distributions, namely, beta (⧫), uniform (■), exponential (▲), and power-law (●). The unfilled symbols represent high degree search for beta (∗), uniform (○), exponential (+), and power-law (◇). Notice that the scaling of high-degree search is same irrespective of the edge-weight distribution. Whereas, the scaling of LBC search decreases as the the heterogeneity in edge-weight increases. This implies that when the heterogeneity in edge-weights is high, the LBC search utilizes low weight edges for navigation.

neighbor i is given by

$$L(i) = (k_i - 1)(n - 2) + (k_i - 1)(n - k_i),$$

where k_i is the degree of the neighbor. The first term is due to the shortest paths from $k_i - 1$ neighbors (j) of node i to $n - 2$ remaining nodes (other than node i and the neighbor j) in the network. The second term is due to the shortest paths from $n - k_i$ nodes (other than $k_i - 1$ neighbors and node i) to $k_i - 1$ neighbors of node i. Note that we chose not to explicitly take into account of the symmetry of distance in undirected networks and count the s-t and t-s paths separately. $L(i)$ is an increasing function if $k_i < n - \frac{1}{2}$, a condition that is always satisfied since $n = 1 + \sum_{j=1}^{d} k_j$. This implies that in a local network with tree-like structure, the

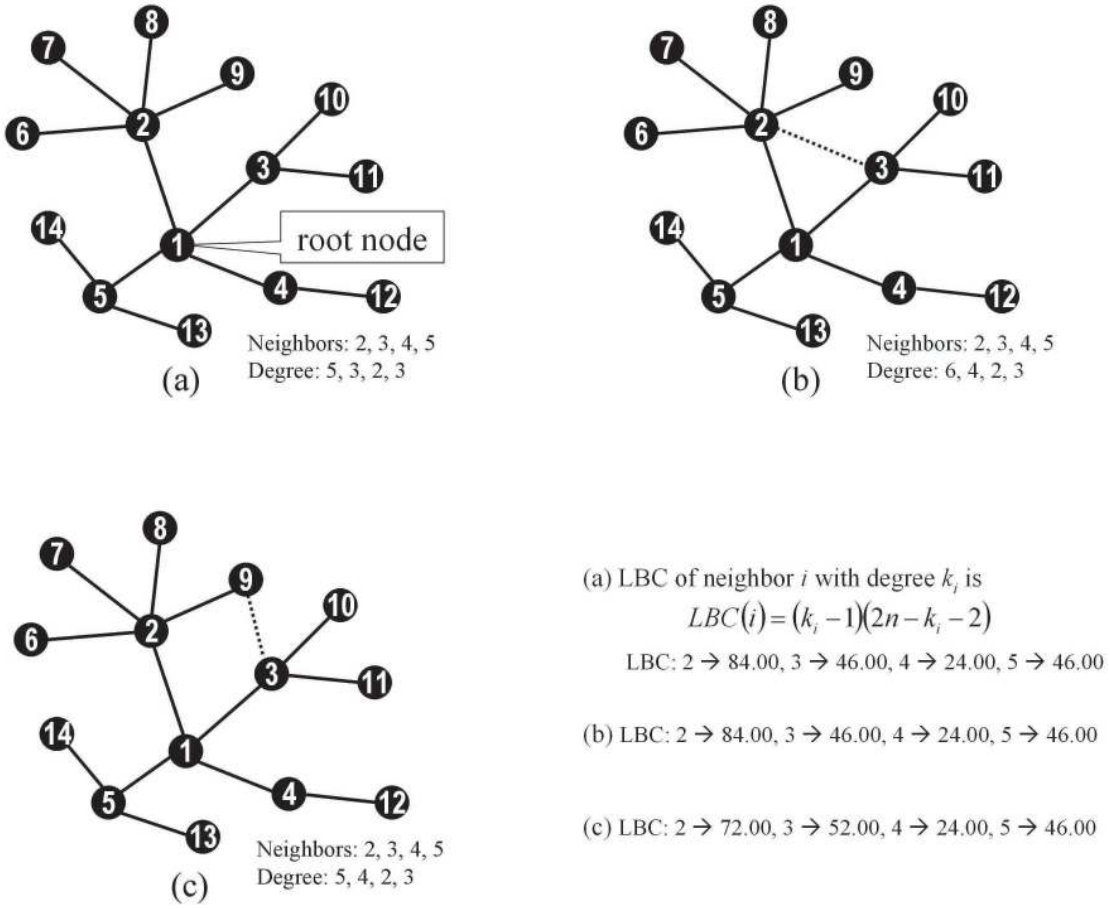


Figure 4.5. (a) A configuration of a local network with a tree like structure. In such local networks, the neighbor with the highest degree has the highest LBC. (b) A local network with an edge between two first neighbors. Here again the neighbor with the highest degree has the highest LBC. (c) A local network with an edge between a first neighbor and a second neighbor. Although there is change in LBCs of neighbors, the order remains the same.

neighbor with highest degree has the highest LBC. We extend the above result for other configurations of the local network by considering different possible cases.

The possible edges other than the edges present in a tree-like local network are an edge between two first neighbors, an edge between a first neighbor and a second neighbor and an edge between two second neighbors. We show that adding any of these two types of edges to a local network with a tree-like structure will not affect the rank order of the neighbors' LBCs. As shown in figure 4.5(b), an edge among two first neighbors changes the LBC of the root node but not that

of the neighbors. Figure 4.5(c) shows a configuration of a local network with an edge added between a first and a second neighbor. Now, there is a small change in the LBCs of the neighbors (nodes 2 and 3) which are connected to a common second neighbor (node 9). Since node 9 is now shared by neighbors 2 and 3, the LBC contributed by node 9 is divided between these two neighbors. The LBC of such a neighbor i is

$$L(i) = (k_i - 2)(n - 2) + (k_i - 2)(n - k_i) + (n - k_j - 1),$$

where k_i is the degree of the neighbor i and k_j is the degree of the neighbor with which node i has a common second neighbor.. The decrease in the LBC of neighbor i is $(n - k_i + k_j - 1)$. If there are two neighbors with the same degree (one with a common second neighbor and another without any) then the neighbor without any common second neighbors will have higher LBC. Another possible change of order with respect to LBC would be with a neighbor l of degree $k_l = k_i - 1$ (if it exists). However, $L(i) - L(l) = (n - k_i - k_j + 1)$ is always greater than 0, since $n = 1 + \sum_{j=1}^{d} k_j$. Thus the only scenario under which the order of neighbors with respect to LBC is different than their order with respect to degree when adding an edge between first and second neighbors is if that creates two first neighbors with the same degree. A similar argument leads to an identical conclusion in the case of adding an edge between two second neighbors as well.

The above discussion suggests that the highest degree neighbor is always the same as the highest LBC neighbor. This is not true in few peculiar instances of local networks. For example, consider the network shown in figure 4.6 which has several edges between the first and second neighbors. We see that the highest degree neighbor is not the same as the highest LBC neighbor. In this local network, the highest degree first neighbor (node 2), participates in several four-node circuits that include the root node. Thus, there are multiple shortest paths starting from second-neighbor nodes on these cycles (nodes 6, 7, 9, 10) and the contributions to node 2's LBC from the paths that pass through it are smaller than unity, consequently the LBC of node 2 will be relatively small. This may be one of the reasons why the highest-degree neighbor node 2 is not the highest LBC neighbor. We feel that this happens only in some special instances of local networks. From

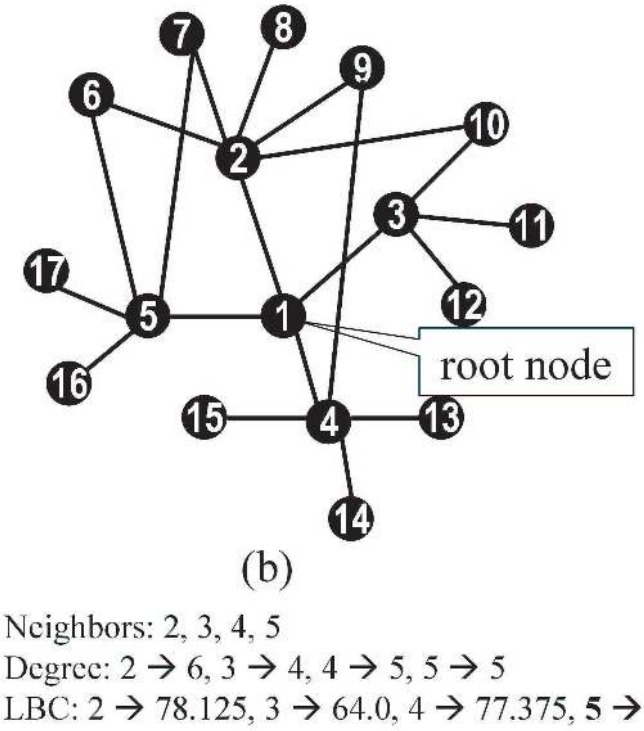


Figure 4.6. An instance of a local network where the order of neighbors with respect to LBC is not same as the order with respect to node degree.

about 50,000 simulations across different types of power-law networks we found that in 99.63 % of cases the highest degree neighbor is the same as the highest LBC neighbor. Hence, we can conclude that in un-weighted networks the neighbor with highest LBC is usually identical to the neighbor with the highest degree.

4.3.3 Optimal neighborhood length for LBC search

In this section, we investigate the effect on the performance of the LBC search, of computing the most central neighbor (Highest LBC neighbor) within a larger local neighborhood. Previously, we calculated the highest LBC neighbor in the local network of neighborhood length 2 (i.e. in the local network consisting of first neighbors and second neighbors). Intuitively, one may feel that if we compute the most central in a larger neighborhood, the performance of the search algorithm may improve. Figure 4.7 shows the local networks formed for different neighborhood lengths. If we consider the whole network for computing the LBC

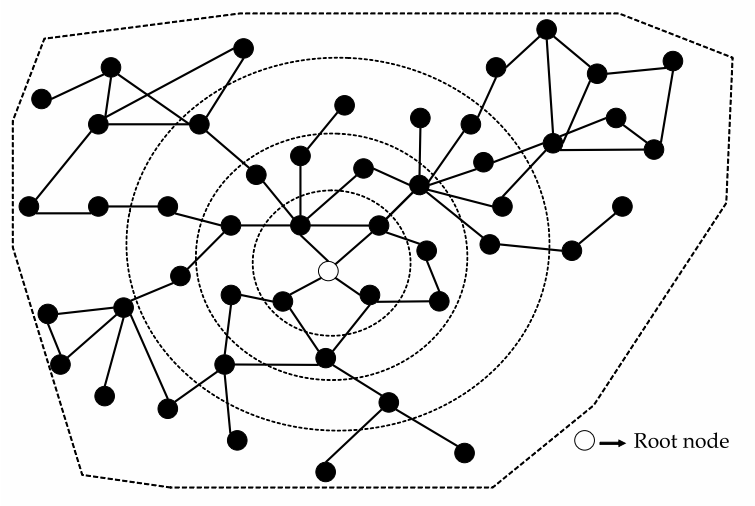


Figure 4.7. Illustration of a network with neighborhood depths of different length. The inner most circle consists of the local network of neighborhood depth 1. Similarly, the next circles show the local networks of depth 2 and 3 respectively. The network in the outer most circle is whole network used for calculating the BC of neighbors.

then it would be same as BC, which gives most central neighbor with respect to the global information of the network. We simulated the LBC search algorithm for different neighborhood lengths and denote it as high LBC_i search, where i is the neighborhood length. If we consider the whole network for calculating the most central neighbor then we call it as high BC search. Table 4.5 compares the results obtained for LBC search with different neighborhood lengths. We observe that LBC_1 performs similar to the random walk search. This is not surprising because if we consider the local network of neighborhood 1, the LBC value for all the neighbors is the same and hence one of the neighbors is chosen randomly. Surprisingly, on the other hand, we notice that the performance of LBC_2, LBC_3, and BC is similar. This implies that LBC_2 search tends to choose the highest BC neighbor even though it is computed using the local network with neighborhood length of 2. This observation strengthens our claim in section 4.3.1 that there exists a positive scaling relationship between local and global betweenness centrality. Further, it demonstrates that LBC search performs as well as BC based search which considers the maximum information available about a nodes first neighbors.

Table 4.5. Comparison of different search algorithms in power-law networks of size 1000 nodes for different edge weight distributions. The mean for all the edge weight distributions is 5 and the variance is σ^2. LBC_i denotes the LBC search algorithm with highest LBC neighbor calculated based on the local networks of depth i. The BC search sends the message to the most central (highest BC) neighbor. The values in the table are the average distances obtained for each search algorithm in these networks. We observe that LBC_1 performs similar to the random walk search. On the other hand, we notice that the performance of LBC_2, LBC_3, and BC is similar.

Search algorithm	Beta $\sigma^2 = 2.3$	Uniform $\sigma^2 = 8.3$	Exp. $\sigma^2 = 25$	Power-law $\sigma^2 = 4653.8$
Random walk	732.8	726.2	725.5	747.7
LBC_1	722.5	688.2	666.1	608.2
LBC_2	275.5	247.4	231.6	195.4
LBC_3	269.7	243.4	228.1	188.0
BC	269.6	240.6	228.9	196.1

4.4 Search in Gnutella

As discussed earlier in section 3.4, Gnutella is a decentralized and unstructured peer-to-peer network used for sharing information between different users. It does not have any centralized server which will index all the users (represented as nodes) and files available. Each node is connected to few other nodes and has information about the files available with the neighbors. If the files are not available with the neighbors, they are searched by sending a query to the neighboring nodes. We obtained the data of Gnutella from Lada Adamic [6] which consists of 574 nodes and 832 edges. The degree distribution of the network follows a power-law $p(k) \sim k^{-\gamma}$ with exponent $\gamma = 2.4 \pm 0.1$ (see figure 4.8). We use this network to investigate the performance of the search algorithms on a real-world network. However, the information regarding the edge-weights of the Gnutella network is not available. Hence, we again generate the edge-weights from different distributions like Beta, uniform, exponential and power-law.

Table 4.6 compares the performance of the search algorithms on Gnutella with different edge-weight distributions. Surprisingly, we notice that minimum edge weight algorithm performs the best except for Beta distribution. The performance of high degree search is similar to LBC search. However, if we simulate the search algorithms on a random network with same numbers of nodes, edges, and power-

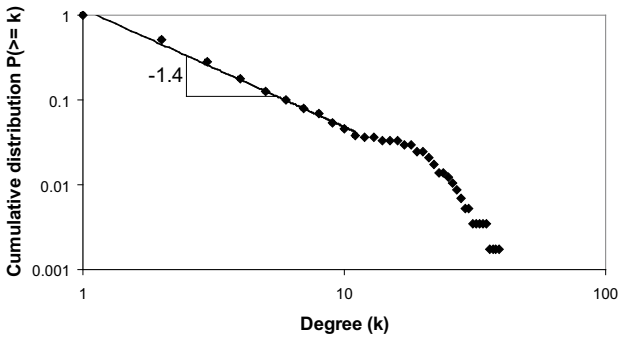


Figure 4.8. Cumulative degree distribution of the Gnutella network. It follows a power-law with exponent 2.4.

law exponent, we find that LBC search performs the best (see table 4.7). It is not clear why the results obtained for Gnutella are not consistent with the simulation results. The root cause must be a network feature not incorporated in the degree distribution, probably related to degree-degree correlations or the over-expression of network motifs. Further discussion of this unexpected behavior is detailed in the section on future work.

Table 4.6. Comparison of different search algorithms in the Gnutella network of size 574 with different edge weight distributions. The mean for all the edge weight distributions is 5 and the variance is σ^2. The values in the table are the average distances obtained for each search algorithm in these networks. The values in the brackets show the relative difference between average distance for each algorithm with respect to the average distance obtained by the LBC algorithm. Unlike the power-law random networks, the difference between the LBC search and high degree search is not substantial. Also, surprisingly minimum edge weight search performed well, especially for power-law edge weight distributions

Search algorithm	Beta $\sigma^2 = 2.3$	Uniform $\sigma^2 = 8.3$	Exp. $\sigma^2 = 25$	Power-law $\sigma^2 = 4653.8$
Random walk	825.79 (40%)	836.90 (38%)	816.81 (43%)	821.92 (46%)
Minimum edge weight	**707.38 (20%)**	**601.38 (-1%)**	**543.83 (-5%)**	**507.81 (-10%)**
Highest degree	614.73 (4%)	624.26 (3%)	604.23 (6%)	583.67 (4%)
Minimum average node weight	803.00 (36%)	674.05 (11%)	588.35 (3%)	531.71 (-5%)
Highest LBC	**590.95**	**605.13**	**571.73**	**561.79**

Table 4.7. Comparison of different search algorithms in power-law networks with exponent, number of nodes, and number of edges same as Gnutella network for different edge weight distributions. The mean for all the edge weight distributions is 5 and the variance is σ^2. The values in the table are the average distances obtained for each search algorithm in these networks. The values in the brackets show the relative difference between average distance for each algorithm with respect to the average distance obtained by the LBC algorithm. LBC search, which reflects both the heterogeneities in edge weights and node degree, performed the best for all edge weight distributions.

Search algorithm	Beta $\sigma^2 = 2.3$	Uniform $\sigma^2 = 8.3$	Exp. $\sigma^2 = 25$	Power-law $\sigma^2 = 4653.8$
Random walk	511.87 (110%)	514.80 (136%)	507.30 (139%)	324.14 (348%)
Minimum edge weight	396.43 (63%)	286.73 (32%)	243.33 (15%)	99.85 (38%)
Highest degree	245.61 (1%)	234.91 (8%)	236.42 (11%)	90.11 (24%)
Minimum average node weight	535.19 (120%)	408.03 (87%)	335.99 (58%)	137.17 (89%)
Highest LBC	**243.67**	**217.75**	**212.16**	**72.41**

CHAPTER

FIVE

Search in spatial networks

In this chapter, we study the decentralized search problem in a family of parameterized spatial network models that are heterogenous in node degree. We investigate several algorithms and illustrate that some of these algorithms exploit the heterogeneity in the network to find short paths by using only local information. In addition, we demonstrate that the spatial network model belongs to a class of *searchable networks* for a wide range of parameter space. Further in section 5.3, we test these algorithms on the U.S. airline network which belongs to this class of networks and demonstrate that searchability is a generic property of the U.S. airline network. These results provide insights on designing the structure of distributed networks that need effective decentralized search algorithms.

5.1 Decentralized algorithms

A simple search algorithm in spatial networks is *greedy search*, where each node passes the message to the neighbor closest to the target node. Let d_i be the distance to the target node from each neighbor i (see figure 5.1(a)) and let k_i be the degree of the neighbor i. Greedy search chooses the neighbor with the smallest d_i. This will ensure that the message is always going to the neighbor closest to the target node. However, greedy search may not be optimal in spatial scale-free networks that have high heterogeneity in node degree. Adamic *et al.* [6] and Thadakamalla *et al.*

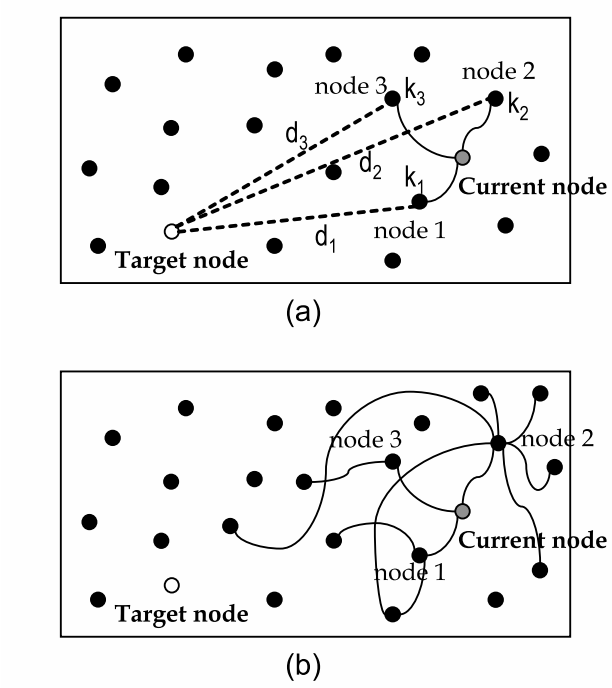


Figure 5.1. (a) Illustration of a spatial network. d_i is the distance to the target node from each neighbor i and k_i is the degree of the neighbor i. (b) Illustration for demonstrating that sometimes it is better to choose a neighbor with higher degree i.e., node 2 over node 1, even if we are going away from the target. This will give higher probability of taking a longer step in the next iteration.

[156] have shown that search algorithms that utilize the heterogeneities present in the network perform substantially better than those that do not. Indeed, choosing a neighbor with higher degree, even by going away from the target node, gives a higher probability of taking a longer step in the next iteration. For instance, in figure 5.1(b), it is better to choose node 2 instead of node 1 since node 2 can take a longer step towards the target node in the next iteration. In the following paragraph, we show that the expected distance a neighbor can take in the next iteration is a strictly increasing function of its degree.

We define the length of an edge as the Euclidian distance between the two nodes connected by the edge. Let $P(X)$ be the probability distribution of edge lengths. Let $Y_k = Max\{X_1, X_2, X_3, \cdots, X_k\}$, where $X_1, X_2, X_3, \cdots, X_k$ are independent and identically distributed (i.i.d.) random variables with distribution function

$P(X)$. The cumulative distribution function of Y_k is

$$P[Y_k \leq y] = \prod_{i=1}^{k} P[X_i \leq y] = [P(X_1 \leq y)]^k$$

This implies

$$E(Y_k) = \int_0^\infty (1 - [P(X_1 \leq y)]^k) dy.$$

Since $P(X_1 \leq y) \leq 1 \ \forall y$,

$$[P(X_1 \leq y)]^{k_1} \leq [P(X_1 \leq y)]^{k_2} \ if \ k_1 \geq k_2,$$

implying that

$$E(Y_{k_1}) \leq E(Y_{k_2}) \ \forall y \text{ if } k_1 \leq k_2$$

Similarly, we can show that if $P(X)$ is not a delta function then

$$E(Y_{k_1}) < E(Y_{k_2}) \text{ if } k_1 < k_2.$$

Now consider two neighbors n_1 and n_2 with degree k_1 and k_2. The expected distance the neighbors n_1 and n_2 can take in the next iteration irrespective of the direction is given by $E[Y_{k_1-1}]$ and $E[Y_{k_2-1}]$ respectively. This implies that $E[Y_{k_1-1}] > E[Y_{k_2-1}]$ if $k_1 > k_2$. Here, we approximate that $X_1, X_2, X_3, \cdots, X_k$ are independent which is valid when the number of edges is large. Hence, if we choose a neighbor with higher degree then there is a greater probability of taking a longer step in the next iteration. Thus one expects that in spatial scale-free networks the efficient algorithm should combine the direction of travel, quantified by d_i, and the degree of the neighbor, k_i, into one measure. Since the units of d_i and k_i are different, there is no trivial way of composition that is optimal. The aim of the measure is to choose a neighbor with smaller d_i and larger k_i with an intuition that a higher degree node should effectively decrease the distance from the target – a goal which can be achieved in many different ways. One could give an incentive $g(k_i)$, and then subtract it from the distance d_i; one could also divide d_i either by k_i or by any increasing function of k_i. We investigated the following search algorithms, which cover a broad spectrum of possibilities:

1. *Random walk*: The node attempts to reach the target by passing the message to a randomly selected neighbor.

2. *High degree search*: The node passes the message to the neighbor with the highest degree. The idea here is that by choosing a neighbor that is well-connected, there is a higher probability of reaching the target node. Note that this algorithm requires the fewest number of hops to reach the target in non-spatial networks [6].

3. *Greedy search*: The node passes the message to the neighbor i with the smallest d_i. This will ensure that the message is always going to the neighbor closest to the target node.

4. *Algorithm 4*: The node passes the message to the neighbor i with the smallest measure $d_i - g(k_i)$. The function $g(k_i)$ is an incentive for choosing a neighbor of higher degree. Ideally, $g(k_i)$ should be the expected maximum length of an edge from a node with degree k_i.

5. *Algorithm 5*: The node passes the message to the neighbor i that has the smallest measure $(\frac{d_i}{d_m})^{k_i}$, where d_m is the Euclidian distance between the most spatially distant nodes in the network, and is used for normalizing d_i. We assume that d_m is known to all the nodes in the network. Note that the algorithm prefers the neighbor that has lower d_i and higher k_i.

6. *Algorithm 6*: The node passes the message to the neighbor i that has the smallest measure $\frac{d_i}{k_i}$. Here, again, the algorithm prefers the neighbor that has lower d_i and higher k_i.

7. *Algorithm 7*: The node passes the message to the neighbor i that has the smallest measure $(\frac{d_i}{d_m})^{\ln k_i + 1}$. This is a conservative version of algorithm 5 with respect to k_i.

8. *Algorithm 8*: The node passes the message to the neighbor i that has the smallest measure $\frac{d_i}{\ln k_i + 1}$. This algorithm is weaker version of algorithm 6 with respect to k_i.

Algorithms from 4 to 8 aim to capture both the direction of travel and the neighbors' degree. Thus, we expect these algorithms to give smaller path lengths than other algorithms. In case of algorithm 4, it would be extremely difficult to define a function independent of the parameters of the network. Hence, it may not be realistic to use this form of composition for direction of travel and degree of neighbor. Even greedy search has a slight preference for high degree nodes, since the probability of reaching a node with degree k is $\sim kp_k$ [125], where p_k is the fraction of nodes with degree k. Hence, the proposed algorithms have to be extremely competitive to perform better than greedy search. The algorithms described above are mainly based on intuition. However, as we discuss later in this chapter, the successful strategies are not restricted to these functional forms.

5.2 Spatial network models: Simulation and Analysis

We investigate the search algorithms by simulating them on the networks generated by the spatial network model detailed in chapter 3. The network is generated on a two dimensional grid with length $a = 1000$, breadth $b = 500$, and $m = 1$ for different values of N, p, and different functions F. Once the network is formed, we randomly choose K pairs (source and target) of nodes and simulate the search algorithms. The source, and consecutively each node receiving the message, passes the message to one of its neighbors, according to the search algorithm. For algorithm 4, we assume the incentive function $g(k_i)$ to be the expected maximum distance a node with degree k_i can take for the next hop, that is, the expected maximum length of an edge from a node with degree k_i. Empirically we found that this function follows the form $c_1 * \ln k_i + c_2$ for all the spatial networks. For algorithms 5 and 7, we let d_m be $\sqrt{a^2 + b^2}$, the largest distance between two points in the considered space. We assume that it is sufficient if the message reaches a small neighborhood of the target node defined by a circle with radius D. This is a realistic assumption in many real-world networks, e.g. it is sufficient if we reach one of the airports in the close neighborhood of a destination city (especially when the city has multiple airports). The search process continues until the message

reaches a neighbor of the target node or a node within a circle of radius $D = 50$ centered around the target node. In order to avoid passing the message to a neighbor that has already received the message, a list L is maintained. During the search process, if the message reaches a node i whose neighbors are all in the list L, then the message is passed to one of the neighbors using the same algorithm. In the case of random walk or high degree search, the message is routed back to the previous node and this particular neighbor i is marked to note that it cannot pass the message any further. If the number of hops exceeds $N/2$, then the search process stops, noting that the path was not found. For each search algorithm, the average path length, l, measured as the number of edges in the path, the average physical distance traveled along the path, d_{path}, and the percentage of times the search algorithm is unable to find a path, c, are computed from the search results obtained for K pairs in 10 instances of the network model. The lower the value of l, d_{path} and c, the better the performance of the search algorithm. We use the shortest average path length and average physical distance obtained by global breadth-first-search (BFS) algorithm and Dijkstra's algorithm [48] respectively, as a benchmark for comparing the performance of the search algorithms.

Table 5.1 compares the performance of different search algorithms for the spatial network, $G(1000, 0.72, 1, d^r, 2)$ with $r = 1$, 2, and 3. We find that the decentralized search algorithms 5, 6, 7, and 8 perform as well as the shortest path obtained using global information of the network. Specifically, the difference between the shortest path and the path obtained by algorithms 6 and 7 is less than a hop. These results are surprising because the latter algorithms only use the local information in the network, yet they perform as well as the BFS algorithm. This behavior is mainly due to the power-law nature of the spatial network: the few nodes with high degree are allowing the algorithms to make big jumps during the search process (see Table 5.1). This conclusion is corroborated by the fact that an increase in r, meaning a decrease in the power-law regime in the degree distribution [31], induces an increase in the path length. Greedy search which uses only the direction of travel is able to find short paths (compare l's in Table 5.1) but for a few node pairs it is unable to find a path (compare c's in Table 5.1). Greedy search does not consider the degree of the nodes and sometimes the algorithm gets stuck in a loop in sparsely connected regions of the network. In the case of algorithm

4, the composition was not very effective. It is likely that the values of the coefficients, which are difficult to compute, were not optimal. Moreover, the optimal values are highly dependent on the parameters and the configuration of the spatial network. Hence, it would be difficult to generalize the algorithm for all networks and we will not consider it further in our analysis. Random-walk and high-degree search do not consider the direction of travel and hence take an exorbitantly large number of hops. Further, we found that the search algorithms' performance with respect to the path length l and physical distance metric d_{path} was similar. Hence, in the rest of our analysis, we do not discuss these two algorithms and the physical distance metric since the results do not add significant new information.

Table 5.1. Comparison of search algorithms on a spatial scale-free network of 1000 nodes in a two dimensional space with length and breadth equal to 1000 and 500, respectively. l is the average path length for the paths found by the search algorithm, d_{path} is the average physical distance for the paths found by each search algorithm and c is the percentage number of times the path was not found. The table summarizes the average of l, d_{path} and c obtained from 10 simulations of the network with parameters $p = 0.72$ and r for 2000 pairs. Note that the decentralized algorithms 5, 6, 7, and 8 perform as well as the shortest paths found by using global information. Even though the greedy search performs well for the paths found (l and d_{path}), it is sometimes unable to find a path (c).

	$r = 1$			$r = 2$			$r = 3$		
	l	d_{path}	c (%)	l	d_{path}	c (%)	l	d_{path}	c (%)
Random walk	41.68	10957	0	70.47	9414	0	138.07	9024	0
High-degree search	28.35	8032	0	54.85	8805	0	120.15	9848	0
Greedy search	3.37	787	0.17	3.59	600	0.83	4.53	537	2.11
Algorithm 4	10.22	2303	0.12	14.07	1987	0.46	20.08	1806	1.87
Algorithm 5	2.47	646	0	2.97	594	0	4.51	677	0.02
Algorithm 6	2.45	636	0	2.85	565	0	3.73	573	0.02
Algorithm 7	2.54	631	0	2.80	539	0	3.52	527	0.02
Algorithm 8	2.66	646	0	2.87	537	< 0.01	3.54	514	0.07
Shortest path length	2.27	531	NA	2.55	435	NA	3.05	403	NA

Table 5.2. Comparison of search algorithms on spatial scale-free networks with different parameters. l is the average path length for the paths found by each search algorithm and c is the percentage number of times the path was not found. The table summarizes the average of l and c obtained from 10 simulations of the network with parameters N, p, r, and d_{char}. Note that the decentralized algorithms 5, 6, 7, and 8 perform as well as the shortest path found by using global information. Even though the greedy search performs well for the paths found (l), it is sometimes unable to find a path (c).

	$N = 1000$, $r = 1$				$p = 0.72$, $r = 1$				$N = 1000$, $p = 0.72$			
	$p = 0.30$		$p = 0.80$		$N = 500$		$N = 1500$		$d_{char} = 0.5$		$d_{char} = 2.0$	
	l	$c(\%)$	l	$c(\%)$	l	$c(\%)$	l	$c(\%)$	l	$c(\%)$	l	$c(\%)$
Greedy search	6.55	7.93	2.90	0.09	4.09	0.24	3.10	0.44	3.64	0.18	3.92	0.1
Algorithm 5	3.41	0.02	2.35	0	2.83	0	2.40	0	2.46	0.03	2.55	0
Algorithm 6	3.38	0.04	2.38	0	2.81	0	2.38	0	2.49	0	2.59	0
Algorithm 7	3.59	0.19	2.40	0	2.95	0	2.43	0.01	2.66	0.02	2.78	0
Algorithm 8	4.12	0.73	2.49	< 0.01	3.16	< 0.01	2.54	0	2.79	0.04	3.01	0.01
Shortest path length	2.91	NA	2.16	NA	2.30	NA	2.26	NA	2.23	NA	2.23	NA

Similar results are obtained for a wide range of parameters for the spatial network model. Table 5.2 summarizes the results for some of these parameter values. This parameter space covers a broad range of power-law networks with different properties. For example, as the value of p changes from 0.3 to 0.8, the power-law exponent of the degree distribution changes from 2.4 to 1.7 (see figure 5.2(a)), which is the usual range of many real-world networks [14, 33, 56, 122]. Hence we can affirm that the spatial network model belongs to a general class of *searchable networks*. Although we have restricted our results to a discussion of two-dimensional spatial networks, it is easy to verify that these results will be valid for higher dimensions. Further, a large number of decentralized search algorithms are efficient. For instance, in algorithm 6 we divide d_i by k_i, whereas in algorithm 8 we divide d_i by $\ln k_i + 1$ which scales logarithmically with k_i. Both algorithms are found to be efficient. This implies that a wide range of functions $f(x)$ that scale between x and $\ln x$ can be used for decentralized search. Hence, we find that the dependence of the search algorithms on the functional forms is weak and the searchability of these networks lies in their heterogeneous structure rather than the functional forms used in the search algorithm.

5.3 Search in the U.S. airline network

Let us consider the U.S. airline network, where nodes are the airports and two nodes are connected by an edge if there is a direct flight from one airport to another. In this network, navigating along an edge from one node to another represents flying from one airport to another. Suppose our objective is to travel from one place to another using the U.S. airline network. In real life, one can obtain a choice of itineraries from the closest airport to the departure location (departure airport) to the closest airport to the destination location (destination airport) using various sources such as travel agents, airline offices or the World Wide Web. These sources have global information about the network and one can choose the itinerary based on different criteria, such as travel fare, number of stopovers, or total time of travel. Now consider a different scenario – one in which we do not have access to the global information of the network, and each airport has only local information. In other words, each airport has information

about the location of the airports it can fly to and how well these neighboring airports are connected (their degree). We do know the location of the departure airport and the destination airport. The objective is to find a path with the fewest stopovers from the departure airport to the destination. From the departure airport, and consecutively from each intermediate airport, we choose to fly to one of its neighbors based on the degree of the neighboring airport, its location and the location of the destination airport. This process continues until we reach the destination airport or any other airport within a small neighborhood of the destination airport. In real life, it is sufficient if we reach one of the airports near the destination airport. For example, it is sufficient to reach LaGuardia Airport (LGA), New York City if the objective is to reach John F. Kennedy International Airport (JFK), New York City. In our study, as a first order approximation we do not consider the type of airline or travel fare as important parameters. Even though this method of travel is unrealistic, it provides insights on the performance of decentralized search algorithms on real world networks.

5.3.1 Properties of the U.S. airline network

The Bureau of Transportation Statistics [2] has a well-documented database on the departure schedule, number of passengers, flight type etc, for all the flights in the United States of America. We considered the data collected for the service class F (scheduled passenger service) flights during the month of January 2006 to form the U.S. airline network. Each airport is represented as a node and a direct flight connection from one airport to another is depicted as a directed edge. We filtered the data to remove the anomalous edges formed due to redirected flights caused by environmental disturbances or random failures. Further, one would expect to have a flight from airport A to airport B if there is one from B to A; but for a small number of instances this was not true. To simplify the analysis, we added edges to make the network un-directed.

After filtering the data, the airline network had 710 nodes and 3414 edges. The number of nodes and edges in the largest connected component (LCC) were 690 and 3412 respectively. The rest of the analysis in the chapter considers only the LCC of the network. Not surprisingly, the properties of the U.S. airline network are

very similar to the properties of the world wide airline network (WWN) [76]. The average path length for the airline network, which is the average minimum number of flights one has to take to go from one airport to any other, is 3.6. The clustering coefficient, which quantifies local order of the network measured in terms of the number of triangles (3-cliques) present, is 0.41. Hence, the U.S. airline network is also a small-world network [169]. The degree distribution of the network follows a power-law $p(k) \sim k^{-\gamma}$ with exponent $\gamma = 1.9 \pm 0.1$ (see figure 5.2(b)), which is close to the exponent of the WWN, 2.0 ± 0.1 [76]. Further, as observed in the WWN, we find that the most connected airports are not necessarily the most central airports. Figure 5.2(c) plots the normalized betweenness centrality (BC) of a node i, $(b_i / < b >)$, where $< b >$ is the average BC of the network, versus its scaled degree $k_i / < k >$, where $< k >$ is the average degree of the network. The geopolitical considerations used to explain this phenomenon in the WWN [73] do not apply to the U.S. airline network, as it belongs to a single country. In fact, this behavior is due to Alaska which contains a significant percentage of the airports (255 of 690, close to 34%) yet only a few (around 6) are connected to airports outside of Alaska. For instance, the BC of Anchorage, Alaska is significantly higher than its degree (see figure 5.2(c)). If we remove the Alaska airports from the network, then we observe better correlation between the degree of a node and its BC (see figure 5.2(d)).

If an area is separated from the U.S. mainland (such as Alaska and Hawaii), then very few airports connect it to the mainland and it may be difficult for search algorithms to capture these connections between the mainland and the other areas. To investigate the effects of this property on the search process, we simulate the algorithms on three different networks, namely, the U.S. airline network, the U.S. airline network without Alaska, and the U.S. mainland airline network without Alaska, Hawaii, Puerto Rico, the U.S. Virgin Islands, and the U.S. Pacific Trust Territories and Possessions (U.S. mainland network). The latter two networks have statistical properties similar to those of the U.S. airline network. The U.S. airline network without Alaska has 459 nodes and 2857 edges with 455 nodes and 2856 edges in the LCC; the U.S. mainland network has 431 nodes and 2729 edges with 427 nodes and 2728 edges in the LCC.

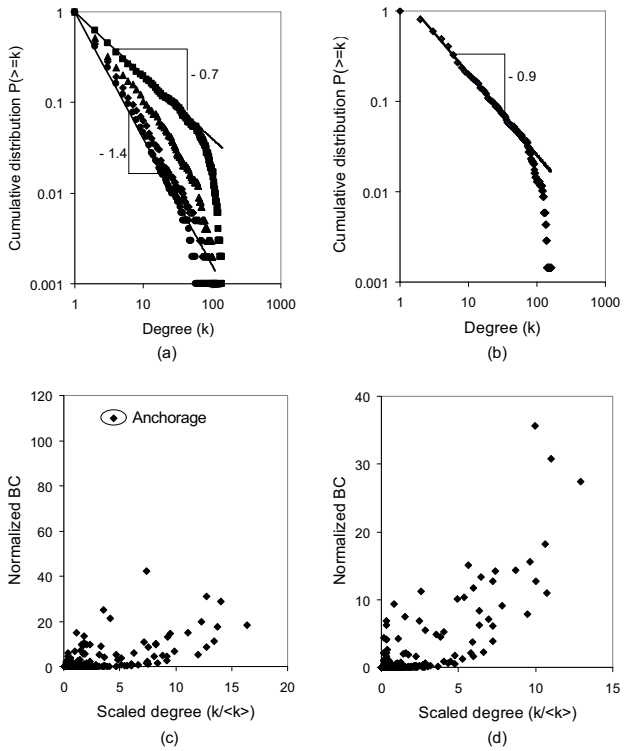


Figure 5.2. (a) Cumulative degree distribution of the networks generated by the spatial network model for different values of p. The symbols represent $p = 0.3$ (•), 0.4 (♦), 0.6 (▲), and 0.8 (■). The power-law exponent of the network can be tuned by changing the value of p. (b) Cumulative degree distribution of the U.S. airline network. (c) Scaling of normalized BC of a node i with its scaled degree for the U.S. airline network. Note that unlike random graphs, there exists no scaling between BC and degree of the node. (d) Scaling of normalized BC of a node i with its scaled degree for the U.S. airline network without Alaska. Note that there is better correlation between BC and degree of the node when compared with the U.S. airline network.

5.3.2 Search results and analysis

We simulated the search algorithms for all $N * (N - 1)$ pairs in each network, where N is the number of nodes. The U.S. airline network, the U.S. airline network without Alaska, and the U.S. mainland network had 475410, 206570, and 181902 pairs, respectively. We chose d_m to be the largest distance between two airports in the network and the neighborhood distance D to be 100 miles. Table 5.3 summarizes the results obtained by each search algorithm. l is the average path length obtained for the paths found by the search algorithm, and c is the number

of times the search algorithm was unable to find a path. The results are similar to the results obtained for the spatial scale-free network model. Algorithms 6, 7, and 8 are able to find paths as short as the paths obtained by the BFS algorithm. Again, greedy search is able to give short paths when it is able to find paths, but there were instances in which it was unable to find any path. In the case of the US airline network without Alaska and the U.S. mainland network, the performance of the search algorithms is even better, especially for algorithm 5 which did not perform well for the complete U.S. airline network. Figure 5.3 visualizes the paths obtained in a characteristic case when greedy search takes a higher number of hops. Often the greedy search reaches the nodes which are near to the destination node but are not well-connected. Hence, it results in traveling many hops within that region before reaching the destination. The proposed search algorithms avoid the low-connected nodes and reach the destination node in fewer hops.

When we looked at the search results in more detail we found a few more interesting behaviors. The greedy search and algorithm 5 were unable to find paths for approximately the same number of pairs in the U.S. airline network (3.54% in the case of the former and 2.92% for the latter). However, there is a difference in the type of paths these search algorithms could not find. The paths not found by greedy search were distributed uniformly for all departure and destination nodes; the paths not found by algorithm 5 were due predominantly to the 18 airports in Alaska, which were unreachable, almost regardless of the starting point. It was interesting to see that even if we start from Anchorage International Airport (ANC), the most connected airport in Alaska, these airports were not reachable. This is mainly due to the high affinity of the algorithm 5 for high degree nodes. The degree of neighbors of ANC which are in Alaska is small compared to the degree of neighbors on the U.S. mainland. Hence, when we start from an airport, the algorithm was able to reach Anchorage but afterward selected one of the highly-connected airports on the U.S. mainland. From that point on, it is difficult to return to Alaska, since the search algorithm is self-avoiding and since the only other airport that flies to Alaska, excluding ANC, is Seattle-Tacoma International Airport (SEA). The U.S. airline network without Alaska and the U.S. mainland network do not have these constraints, and hence algorithm 5 was able to perform better.

Among the 475410 pairs of source and destination nodes searched, algorithms 6 and 7 could not reach the destination node 752 and 688 times, respectively. Again, it turns out that the failure to reach the destination was mainly due to a particular airport, namely, Havre City-County Airport (HVR) in Montana. Similar behavior was observed for these algorithms in the U.S. airline network without Alaska and the U.S. mainland network. HVR is a single-degree node that is connected to Lewistown Airport (LWT), Montana and the only other airport to which LWT is connected is Billings Logan International Airport (BIL), Montana which is a well-connected airport. Hence, the only way to reach HVR would be to reach BIL first and then to fly to LWT. Unfortunately, none of the algorithms, other than the greedy search, can choose LWT from BIL when the destination is HVR. Here again, even though the algorithms 5, 6, 7, and 8 are able to reach BIL, they do not choose LWT as the first choice. Moreover, once they fly out of BIL, they take many hops to reach BIL again due to the self-avoiding nature of the algorithms. For instance, when the destination is HVR, algorithms 7 and 8 take, on an average, only 2.5 and 3.44 hops respectively to reach BIL. However, to reach HVR they take around 170 and 102 hops, respectively. The reason why this behavior is not observed for other single-degree nodes in the U.S. mainland network is that single-degree nodes are usually connected to high degree nodes. The average degree of the neighbors of the single-degree nodes was found to be 82.86, which is significantly higher than the average degree in the network (12.78). In addition, the only airport (LWT) that flies to HVR (or to a neighborhood of HVR) is not chosen by the only other airport (BIL) that can fly to LWT.

Table 5.4 gives the percentage of times the path length found by the search algorithms is the same as the shortest path length. In approximately 90% of the pairs, the path length found by algorithms 6, 7, and 8 was the same as the shortest path length. Further, in 97% of the pairs, the path length found was more than the shortest path by a maximum of two hops. Given that the search algorithms use only local information these results on the airline networks are quite fascinating. Note that this behavior is due mainly to the inherent structure of the U.S. airline network, which can be considered a "searchable network".

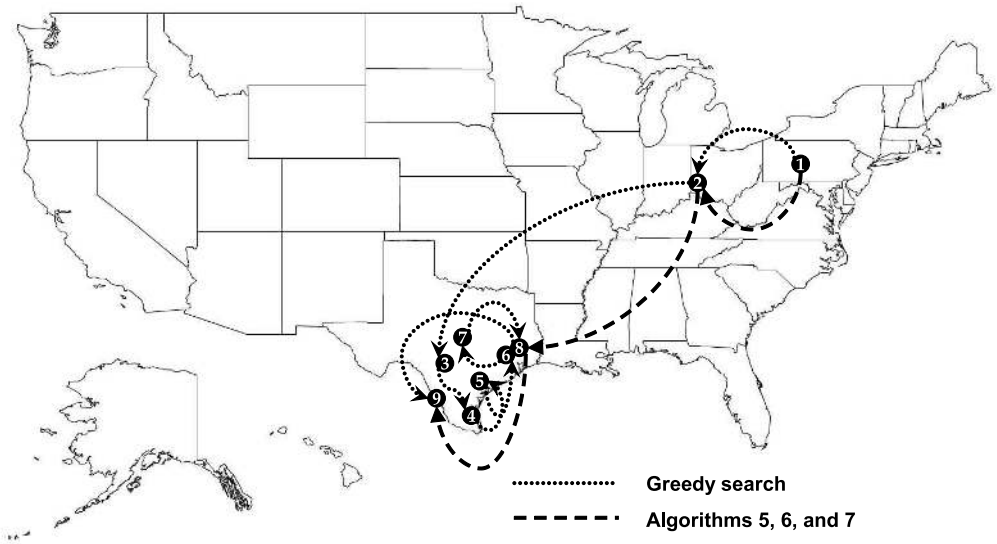


Figure 5.3. Visualization of the paths obtained in a characteristic case when greedy search takes a higher number of hops. In this case, the departure airport is State College, PA (node 1) and the destination airport is Laredo, Texas (node 9). The airline codes and degrees corresponding to the nodes are: 1, SCE, degree 5; 2, CVG, degree 118; 3, SAT, degree 29; 4, HRL, degree 6; 5, CRP, degree 5; 6, HOU, degree 31; 7, AUS, degree 34; 8, IAH, degree 118; 9, LRD, degree 2. The path obtained for the greedy search is $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9$ and for the algorithms 5, 6, and 7 is $1 \rightarrow 2 \rightarrow 8 \rightarrow 9$. Algorithm 8, not shown on the map, takes 4 hops ($1 \rightarrow 2 \rightarrow 3 \rightarrow 8 \rightarrow 9$). Often the greedy search reaches the nodes which are near to the destination node but are not well-connected. Hence, it ends up traveling many hops within that region before it reaches the destination. Whereas, the proposed search algorithms avoid the low-connected nodes and reach the destination node in a lesser number of hops.

Table 5.3. Comparison of search algorithms on the U.S. airline network, the U.S. network without Alaska, and the U.S. mainland network. l is the average path length for the paths found by the search algorithms and c is the number of times the path was not found. The table summarizes the average of l and c obtained for all the possible pairs in the network. In the U.S. airline network, algorithms 6, 7, and 8 give paths close to the shortest path length. In the other two networks, algorithms 5, 6, 7, and 8 give short paths. Here again, the greedy search performs well for the paths found (l) but it is sometimes unable to find a path (c).

	U.S. airline network (N = 690, Pairs = 475410)		U.S. network without Alaska (N = 455, Pairs = 206570)		U.S. mainland network (N = 427, Pairs = 181902)	
	l	c	l	c	l	c
Greedy search	3.93	16806 (3.54%)	2.83	4015 (1.94%)	2.74	3729 (2.05%)
Algorithm 5	5.53	13870 (2.92%)	3.75	456 (0.22%)	2.85	425 (0.23%)
Algorithm 6	4.01	752 (0.16%)	3.17	454 (0.22%)	2.68	425 (0.23%)
Algorithm 7	3.37	688 (0.14%)	2.68	453 (0.22%)	2.93	1 ($<<$ 0.01%)
Algorithm 8	3.37	41 ($<$ 0.01%)	2.76	38 (0.02%)	2.75	39 (0.02%)
Shortest path length	3.02	NA	2.39	NA	2.32	NA

Table 5.4. Comparison of search algorithms on the U.S. airline network, the U.S. network without Alaska, and the U.S. mainland network. "Diff = 0" is the percentage of pairs for which the path length found by the search algorithms is the same as the shortest path length. Algorithms 6, 7, and 8 are able find the shortest paths in more than 90% of the pairs. "Diff $\leq$ 2" is the percentage of pairs for which the path length found was more than the shortest path by a maximum of two hops. Given that the search algorithms use only local information, these results on the U.S. airline network are quite fascinating.

	U.S. airline network		U.S. network without Alaska		U.S. mainland network	
	Diff = 0 (%)	Diff $\leq$ 2 (%)	Diff = 0 (%)	Diff $\leq$ 2 (%)	Diff = 0 (%)	Diff $\leq$ 2 (%)
Greedy search	66.3	85.8	75.3	92.3	75.8	92.7
Algorithm 5	66.9	72.1	88.2	93.7	90.8	96.0
Algorithm 6	88.8	96.6	90.8	95.6	92.2	96.8
Algorithm 7	91.3	98.0	92.0	97.6	92.4	98.1
Algorithm 8	88.4	97.5	89.5	97.8	89.0	97.6

CHAPTER

SIX

Conclusions and Future work

Complex networks abound today's world and are continuously evolving. The sheer size and complexity of these networks pose unique challenges in their design and analysis. Such unordered networks are so pervasive that there is an immediate need to develop new analytical approaches. In this thesis, we presented significant findings and developments in recent years that led to a new field of inter-disciplinary research, *Network Science*. We discussed how network approaches and optimization problems are different in network science than traditional OR algorithms. The fundamental difference is that due to the size of the network with no pre-specified order, these are characterized based on macroscopic properties such as degree distribution and clustering coefficient rather than the individual properties of the nodes and edges. Importantly, these macroscopic or statistical properties have a huge influence on the dynamic processes taking place on the network. Therefore, to optimize a process on a given configuration, it is important to understand the interactions between the macroscopic properties and the process. This will further help in the design of optimal network configurations for various processes. In this thesis, we mainly focused on search and routing, which is the most important and prevalent process in many real-world networks. In specific, we concentrated on search and routing in the network when the available information is limited. We broadly classified the problem of search as search in non-spatial networks and search in spatial networks. The conclusions obtained from the study of decentral-

ized search problem on these two types of networks are summarized below.

6.1 Search and routing in non-spatial networks

For non-spatial networks, we gave a new direction for decentralized search in networks with heterogeneous edge weights. We proposed a decentralized search algorithm based on a new local measure called local betweenness centrality. We studied complex tradeoffs presented by efficient decentralized search and showed that heterogeneity in edge weights has huge impact on search. Moreover, the impact of edge weights on search strategies increases as the heterogeneity of the edge weights increase. We also demonstrated that the search strategy based on LBC utilizes the heterogeneity in both the node degree and edge weight to perform the best in power-law weighted networks. We observed that the performance of LBC search is similar to BC search, which utilizes the maximum information about a neighbor. However, when tested in a peer-to-peer network, Gnutella, the results were not consistent with the results obtained from simulation. The reasons for this behavior are not completely clear. Further investigation of this unexpected behavior is a topic of future work.

Further, we observed that the exponent for the scaling of LBC search with network size decreases as the heterogeneity in edge weights increase. Whereas, the exponent for scaling of high degree search remains the same. This implies that when the heterogeneity in edge-weights is high, the LBC search exploits low weight edges for navigation. Since many studies [17, 18, 28, 29, 30, 38, 57, 65, 70, 72, 76, 89, 100, 105, 106, 120, 126, 129, 137, 139, 174], have shown that there exists large heterogeneity in the capacity and strengths of the interconnections in the real networks, it is important that local search is based on LBC rather than high degree as shown by Adamic et. al. [6]. Further, we demonstrated that in unweighted power-law networks, the neighbor with the highest degree is usually the same as the neighbor with the highest LBC. Hence, our proposed search algorithm based on LBC is universal and is efficient in a larger class of complex networks.

6.2 Search and routing in spatial networks

In this thesis, we studied decentralized search in spatial scale-free networks. We proposed different search algorithms that combine the direction of travel and the degree of the neighbor and illustrated that some of these algorithms can find short paths by using the local information alone. We demonstrated that a family of parameterized spatial network model belongs to a class of searchable networks for a wide range of parameter space. Further, we tested these algorithms on the U.S. airline network. Surprisingly, we found that one can travel from one place to another in fewer than four hops while using only local information. This implies that searchability is a generic property of the U.S. airline network, as is also the case for social networks.

In addition, the spatial network model and the airline network are searchable for a wide range of search algorithms. For example, algorithms 6 and 8 are both able to find short paths in these networks. Hence, any search algorithm with a function $f(x)$ that scales between x and $\ln x$ should give short paths. Moreover, the algorithms can be extended to other power-law networks if we can embed the network in an n-dimensional metric space in which nodes are connected based on the metric distance. The algorithms are relevant to other networks such as the Internet and road networks. As demonstrated in [173], the Internet can be described by the family of spatial network models considered in this thesis and hence we expect that these search algorithms can find short paths in the Internet. However, road networks do not follow a power-law degree distribution. Investigating the algorithms on the dual form of the road networks, which do exhibit scale-free properties [90], is a topic of future work.

We notice that algorithm 8, the most conservative with respect to degree, performs the best in the U.S. airline network. This implies that direction plays the most important role in efficient search, and even slight blending of direction with degree is sufficient to drastically improve the efficiency of search algorithms. In other words, a search algorithm which traverses based on direction and that cautiously avoids low-degree nodes should give short paths. As observed with algorithm 5, sometimes high preference for degree may lead the algorithm to the nodes far away from the destination node. Further, we can conclude that searchability is

a property of the network rather than of the functional forms used for the search algorithm.

The difference between the results obtained on the U.S. airline network and the U.S. mainland network is not significant (especially for algorithms 7 and 8). This implies that the results can probably be extended to the world-wide airline network (WWN) [76] which has a very similar structure to the U.S. airline network. In the U.S. airline network, we have separated areas which are connected to the mainland by only a few airports. Algorithms 7 and 8 are able to capture these connections in order to travel from one separated area to another. The WWN will have many more of these separated areas which are well-connected locally but are sparsely inter-connected. We feel that algorithms 7 and 8 would be able to find short paths in WWN; verification would be subject to the availability of data on the WWN.

Probably, the results obtained for the U.S. airline network are intuitive. For instance, in real life if one is asked to travel with local information, he/she can always find a short path – if not always the shortest path. But the significance of the results lies in capturing this phenomenon/intuition in an algorithm. Definitely, the structure of the network facilitates its searchability. As conjectured by others, the results presented in this thesis support the hypothesis [6, 98] that many real-world networks evolve to inherently facilitate decentralized search. Furthermore, these results provide insights for designing the structure of decentralized networks that need effective search algorithms.

6.3 Uniqueness and significance of the thesis

In this thesis, we consider a fundamentally new approach for design and analysis of complex engineering systems which can be realized as networks. As detailed below, this research is unique in many ways:

1. It addresses the issues due to the increasing scale of many engineering systems. These large-scale networks have become pervasive in the real world and there is an immediate need to develop new analytical approaches that can handle the complexity of these systems. We are one among the first people to apply tools and techniques offered by this approach for engineering

systems. We published a paper [158], studying supply chains as complex networks.

2. Many problems in complex networks are similar to the research issues in traditional OR. Recently, we wrote a book chapter "Complexity and Large-scale networks" [144] that explains the similarities and differences between these two research fields. Further, we addressed the need and opportunity for the OR community to contribute to this fast-growing research field.

3. Traditional routing algorithms assume global information of the network. We are one among the few who consider decentralized algorithms for search and routing. The nodes interact collectively and achieve a desired global objective. These algorithms became extremely pertinent and significant due to new emerging areas such as wireless sensor networks.

4. Even though it is widely argued that complex networks have unequal edge weights, all the previous research on decentralized search have considered equal edge weights. We are the first ones to design local search algorithms for complex networks with heterogeneous edge weights.

5. We demonstrated that a family of parameterized spatial network model belongs to a class of searchable networks for a wide range of parameter space. Many real-world networks such as the Internet [173] and the worldwide airline network [73], can be described by this family of spatial network models. These results provide insights on designing the structure of distributed networks that need effective decentralized search algorithms.

6.4 Future work

The field of Network Science is still in its infancy. The last few years has witnessed an intense amount of activity across different disciplines such as Computer Science, Biology, Mathematics, Sociology, Physics, Political Science etc. However, the tools and techniques developed so far are not sufficient to completely understand and characterize the structure and function of real-world networks. Most of the results so far have been empirical and there is significant need to develop a systematic

approach and a general mathematical framework for modeling and predicting the behavior of these networks. Here we will focus on the discussion of future work in decentralized search. The following are a few potential areas for further study in decentralized search:

6.4.1 Embedding non-spatial networks

We noticed that decentralized search in non-spatial networks gives a path with large distance. This is mainly because during the search process, it is difficult to know whether we are going towards the target node or away from the target node. However, in real-world networks, especially, social networks there does exist some hidden structure which guides the search process to the specific target node. For example, consider the problem of decentralized search for a specific researcher in the acquaintance network of a scientific community. Say, a researcher S would like to send a message to another researcher T using local information and the acquaintances of this network. The researcher S, and subsequently each researcher who receives the message forward the message to one of their acquaintance whom they judged to be closer (than themselves) to the target T. The search process stops when the message reaches one of the acquaintance of the target researcher T. Table 6.1 summarizes the average path length taken by different algorithms for search in scientific collaboration network formed from the papers in the Los Alamos e-print archive with condensed matter speciality. Each author in the paper is considered as a node and two authors are connected by an edge if they coauthor a paper. This network consists of 16726 nodes with 13861 in the largest connected component. We observe that the search algorithms take an astonishingly large number of steps to reach the target node. However, if experimented in the real-world, sufficient studies exist to suggest that one can reach the target node in far less number of hops (note that the average number of hops taken in Milgram's experiment [111] is 6). This is mainly due to the hidden structure present in the network that guides the search process to reach the target node. The nodes in the network would be aware of this structure as in Milgram's experiment. One would wonder what could be the hidden structure of the network that the nodes are aware off. Kleinberg [95] and later Watts *et al.* [168] proposed different models to explain the emergence of

Table 6.1. Comparison of search algorithms in the scientific collaboration network formed from the papers in the Los Alamos e-print archive with condensed matter speciality. The network consists of 16726 nodes with 13861 in the largest connected component. The values in the table are the average number of hops taken by each search algorithm in this network.

Search algorithm	Average number of hops
Random walk	5284.5
Highest degree	4054.8
Highest LBC	3745.0

such phenomenon.

Unfortunately, the model given by Kleinberg [95] is too constrained and represents only a small subset of complex networks. Whereas, in many real-world networks, it may not be possible to divide the nodes into sets of groups in a hierarchy depending on the properties of the nodes as in the Watts *et al.* model [168]. We need a generalized approach that can capture the hidden structure of the network. Recently, it was proposed that many real networks could be embedded in an Euclidean space of low dimension (see figure 6.1). In Euclidean space, the distance between the nodes would be proportional to the dissimilarities between the nodes. Since there is higher probability of similar nodes being connected, in Euclidean space, the nodes have higher probability of being connected if they are closer to each other. This implies that the embedded network would be similar to the spatial network generated from the model described in chapter 3. Also, once we embed the network, we have a metric which tells us whether we are going towards the target node or away from the target node during the search process. Hence, we strongly feel that the proposed search algorithms would result in short paths for the network embedded in Euclidean space. Navigating in the actual network knowing the hidden structure would be similar to the navigating in the embedded network using the distance metric. Demonstrating this phenomenon would help us understand the hidden structure of the networks and why they are searchable. Further, these results could be applied to vast research areas from designing decentralized wireless sensor networks to understanding the information retrieval process in the spatial brain networks [33].

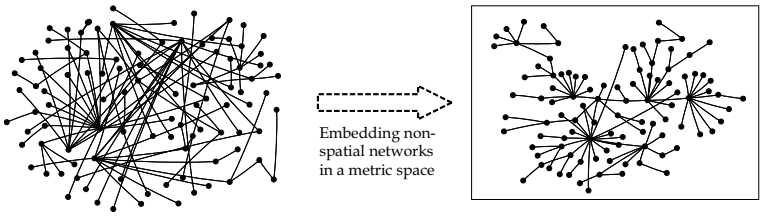


Figure 6.1. Illustration for embedding a non-spatial network in a metric space. (a) Non-spatial network. There could be an hidden structure in the network which the nodes are aware of during the search process. (b) Non-spatial network embedded in a metric space (2-dimensional Euclidian space). Here a distance metric is defined which can guide us during the search process. Navigating in the actual network knowing the hidden structure would be similar to the navigating in the embedded network using the distance metric.

6.4.2 Behavior of Gnutella

In section 4.4, we observed that the performance of the search algorithms in Gnutella network is inconsistent with the results obtained by simulation on random power-law networks (see tables 4.6 and 4.7). In specific, we notice that search algorithms that utilize the high-degree nodes in the network did not perform well. This observation is particularly surprising because the properties of the simulated network is same as the Gnutella with respect to the number of nodes, edges, and power-law exponent of the degree distribution. Even though the clustering co-efficient of Gnutella is higher than the simulated network, it is not clear its influence on high degree affinity search algorithms. Further, other properties such as degree correlations (assortative vs. disassortative) are also found to be similar for both the simulated and the Gnutella network. The Pearson correlation coefficient [121] for both the networks is close to 0.25. If we observe closely, the performance of the search algorithms in Gnutella is similar to the performance of the algorithms in Poisson random network (compare tables 4.2 and 4.6). This implies that even though Gnutella has power-law degree distribution, the heterogeneity in node degree did not help high degree and high LBC search to perform better. Possibly, there are not sufficient high degree nodes due to exponential cutoff observed in the degree distribution of Gnutella (see figure 4.8). Further investigation of this interesting behavior is a good topic of future research. At the same time, we would like to note that the weights on the edges of the Gnutella for the above results are

simulated. Testing the algorithms on the Gnutella with actual edge-weights may lead to different behaviors.

6.4.3 Analytical results

Most of the results in decentralized search have been empirical. To strengthen the claims obtained from these results, we need analytical bounds on the search algorithms. Even though Kleinberg provided analytical bounds for a class of complex network models, they are too constrained and represent only a small subset of complex networks. Most of the study on decentralized search did not address the issue of proving the analytical bounds for the efficiency of the algorithms. This is a promising area of research where there is much to be done. So far, there do not exist a rigorous mathematical framework for analyzing these complex networks and hence it was a difficult task to obtain analytical bounds. However, as Network Science becomes a more mature field, it will offer a lot more promising tools and techniques.

6.4.4 Extension to road networks

Road networks can be represented as a network where intersections are represented as nodes and road segments as the edges (see figure 6.2). This representation leads to a network with homogenous degree distribution since most of the nodes will have degree 4. Now consider the problem of routing from one intersection to another using the road network. When global information is available, one can always calculate the shortest path from one intersection to another with respect to either time or distance. Many services such as Google Maps or MapQuest do the same to provide the directions for the optimal path. However, if the information available is only local, such services may not be useful. Decentralized routing problem in the road networks is to go from one intersection to another along the segments of the road using local information. One simple algorithm is greedy search where the algorithm always choose to go the intersection which is closet to the destination. However, as demonstrated for spatial networks this algorithm may not be optimal and the algorithms proposed in this thesis may perform better in road networks as well. Unfortunately, road networks are not heterogenous in node degree and hence

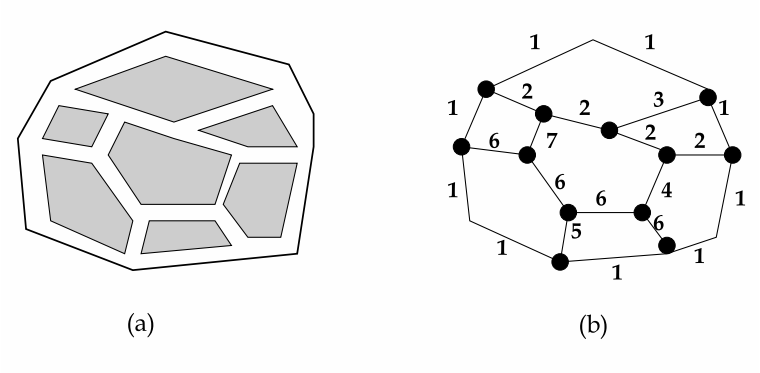


Figure 6.2. Illustration for the graphical representation of a road network on a geographical area. (a) Structural representation of the geographical area consisting of the road network. (b) Graphical representation of the road network in its primal form. The intersections are represented as nodes and road segments as the edges. The numbers on the road segments represent the name of the road.

the algorithms are not directly applicable.

Recently, there has been a body of research [90, 147] that considers road networks in dual form, which do exhibit scale-free properties. In the dual form, a node represents a single road of a given name, and two nodes are connected if the corresponding roads ever meet at an intersection (see figure 6.3). The degree distributions of the road networks in the dual form for United States, England, and Denmark are found to follow power-law, with exponents between 2.2 and 2.4 [90]. This heterogeneity if due to few roads (national highways) which span across large area and have high degree and large number of roads (local streets) that have low degree. Although the algorithms proposed for spatial network are applicable for the dual form of the network, many questions have to be addressed before implementation. For instance, one question would be how to represent the destination point in the dual form of the network? A good methodology would be to navigate in the primal form of the network using the local information from the dual network. We strongly feel that this is a promising area for future research.

6.4.5 Heterogenous wireless sensor networks

As discussed in chapter 3, wireless sensor networks (WSN) promise to revolutionize sensing in wide range of applications. Some of the possible applications are ana-

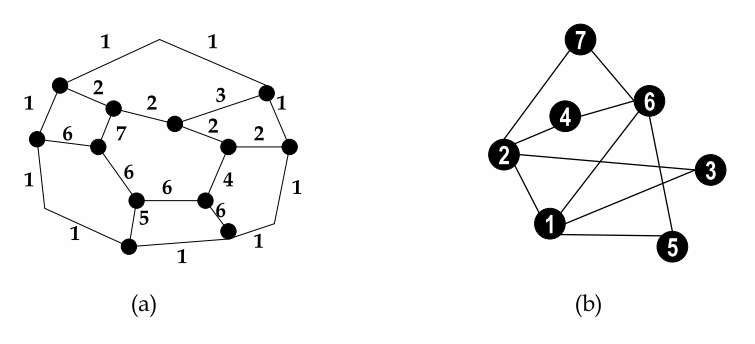


Figure 6.3. Illustration for the primal and dual representation of a road network. (a) Graphical representation of the road network in its primal form. The intersections are represented as nodes and road segments as the edges. The numbers on the road segments represent the name of the road. (b) Dual representation of the road network. A node represents a single road of a given name, and two nodes are connected if the corresponding roads ever meet at an intersection

lyzing the movement of tornadoes, detect forest fires at early stages, alert border guards to activity in remote areas, increase alertness to potential terrorist attacks etc. Early research on WSNs are focused on designing network with sensors of homogenous battery power and equal capabilities. Main advantage of such homogenous design is resilience to individual failures. However, the life-time and reliability of the WSN is found to be highly constrained if we use homogenous sensors [172]. The distribution of resources in many real networks is not homogenous. It has been shown that heterogeneity in the system gives rise to an optimal configuration with respect to many properties such as robustness, routing etc.

Recently, it was demonstrated that using heterogenous sensors, i.e. sensors with different battery powers would significantly improve the life-time of the sensor network [110, 172]. This configuration would have few sensors with a large amount of battery power and large number of sensors with small battery power in WSNs (see figure 6.4). Mhatre *et al.* [110] considered a heterogenous sensor network with two types of sensor nodes; one type is deployed with intensity λ_0, another type that has higher energy and communication capability with intensity λ_1. They demonstrated that the lifetime of the sensor network is maximized when λ_1 scales with the square root of λ_0. Similarly, different other configurations could be considered for heterogenous sensor networks. In fact, the presence of hetero-

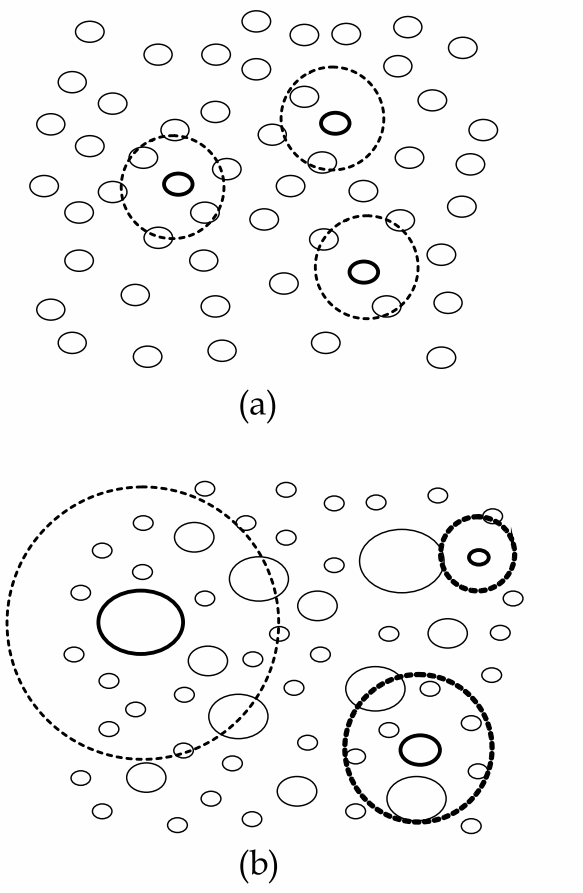


Figure 6.4. Illustration for homogenous and heterogenous wireless sensor networks (a) Wireless sensor network with homogenous power capacity. (b) Wireless sensor network with heterogeneous power capacity. The size of the node represents the amount of battery power and communication capabilities.

geneity in the sensor network will lead to a new class of analytical problems [172]. This heterogeneity could also be utilized for more efficient routing. In this thesis, we demonstrated that the performance of search and routing algorithms is significantly improved when they exploit the heterogeneity present in the network. Hence, we believe that these algorithms could be easily extended for heterogenous wireless sensor networks as well.

BIBLIOGRAPHY

[1] The Internet Movie Database can be found on the WWW at http://www.imdb.com/.

[2] The Bureau of Transportation Statistics, http://www.transtats.bts.gov/, date accessed: July 20, 2006.

[3] J. Abello, P. M. Pardalos, and M. G. C. Resende. *External Memory Algorithms: DIMACS series in discrete mathematics and theoretical computer science*, volume 50, chapter On maximum clique problems in very large graphs, pages 119–130. American Mathematical Society, 1999.

[4] J. Abello and J. Vitter, editors. *External Memory Algorithms: DIMACS series in discrete mathematics and theoretical computer science*, volume 50. American Mathematical Society, Boston, MA, USA, 1999.

[5] L. A. Adamic and B. A. Huberman. Growth dynamics of the world-wide web. *Nature*, 401(6749):131, 1999.

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

[7] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network flows: Theory, Algorithms, and Applications*. Prentice-Hall, NJ, 1993.

[8] W. Aiello, F. Chung, and L. Lu. A random graph model for massive graphs. *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 171–180, 2000.

[9] W. Aiello, F. Chung, and L. Lu. A random graph model for power law graphs. *Experimental Mathematics*, 10(1):53–66, 2001.

[10] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: A survey. *Computer Networks*, 38(4):393–422, 2002.

[11] J. N. Al-Karaki and A. E. Kamal. Routing techniques in wireless sensor networks: a survey. *IEEE Wireless Communications*, 11(6):6–28, 2004.

[12] R. Albert, I. Albert, and G. L. Nakarado. Structural vulnerability of the north american power grid. *Phys. Rev. E*, 69(2):025103, 2004.

[13] R. Albert and A. L. Barabási. Topology of evolving networks: Local events and universality. *Phys. Rev. Lett.*, 85(24):5234–5237, 2000.

[14] R. Albert and A. L. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97, 2002.

[15] R. Albert, H. Jeong, and A. L. Barabási. Diameter of the world wide web. *Nature*, 401(6749):130–131, 1999.

[16] R. Albert, H. Jeong, and A. L. Barabási. Attack and error tolerance of complex networks. *Nature*, 406(6794):378–382, 2000.

[17] E. Almaas, B. Kovacs, T. Viscek, Z. N. Oltval, and A. L. Barabási. Global organization of metabolic fluxes in the bacterium escherichia coli. *Nature*, 427(6977):839–843, 2004.

[18] E. Almaas, P. Krapivsky, and S. Redner. Statistics of weighted treelike networks. *Phys. Rev. E*, 71(3), 2005. art. no. 036124.

[19] R. B. Almeida and V. A. F. Almeida. A community-aware search engine. In Proceedings of the 13th International Conference on World Wide Web, ACM Press, 2004.

[20] L. A. N. Amaral, A. Scala, M. Barthelemy, and H. E. Stanley. Classes of small-world networks. *Proc. Natl. Acad. Sci.*, 97(21):11149–11152, 2000.

[21] C. Anderson, S. Wasserman, and B. Crouch. A p* primer: Logit models for social networks. *Social Networks*, 21(1):37–66, 1999.

[22] R. M. Anderson and R. M. May. *Infectious Diseases in Humans*. Oxford University Press, Oxford, 1992.

[23] A. Arenas, A. Cabrales, A. Diaz-Guilera, R. Guimera, and F. Vega. *Statistical mechanics of complex networks*, chapter Search and Congestion in Complex Networks, pages 175–194. Springer-Verlag, Berlin, Germany, 2003.

[24] G. Bagler. Analysis of the airport network of india as a complex weighted network. 2004. e-print cond-mat/0409773, http://lanl.arxiv.org/abs/cond-mat?papernum=0409773.

[25] J. Balthrop, S. Forrest, M. E. J. Newman, and M. M. Williamson. Technological networks and the spread of computer viruses. *Science*, 304(5670):527–529, 2004.

[26] A. L Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[27] A. L. Barabási, H. Jeong, Z. Neda, E. Ravasz, A. Schubert, and T. Vicsek. Evolution of the social network of scientific collaborations. *Physica A*, 311:590–614, 2002.

[28] A. Barrat, M. Barthelemy, R. Pastor-Satorras, and A. Vespignani. The architecture of complex weighted networks. *Proc. Natl. Acad. Sci.*, 101(11):3747, 2004.

[29] A. Barrat, M. Barthelemy, and A. Vespignani. Modeling the evolution of weighted networks. *Phys. Rev. E*, 70(6):066149, 2004.

[30] A. Barrat, M. Barthelemy, and A. Vespignani. Weighted evolving networks: Coupling topology and weight dynamics. *Phys. Rev. Lett.*, 92(22):228701, 2004.

[31] M. Barthélemy. Crossover from scale-free to spatial networks. *Europhys. Lett.*, 63(6):915–921, 2003.

[32] M. Barthelemy, A. Barrat, R. Pastor-Satorras, and A. Vespignani. Characterization and modeling of weighted networks. *Physica A*, 346:34–43, 2005.

[33] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D. U. Hwang. Complex networks: Structure and dynamics. *Physics Reports*, 424:175–308, 2006.

[34] V. Boginski, S. Butenko, and P. Pardalos. Statistical analysis of financial networks. *Computational Statistics & Data Analysis*, 48:431–443, 2005.

[35] V. Boginski, S. Butenko, and P. Pardalos. Mining market data: a network approach. *Computers & Operations Research*, 33:3171–3184, 2006.

[36] B. Bollobas. *Randon graphs*. Academic, London, 1985.

[37] B. Bollobas and O. Riordan. *Handbook of Graphs and Networks*, chapter Mathematical results on scale-free graphs. Wiley-VCH, Berlin, 2003.

[38] L. A. Braunstein, S. V. Buldyrev, R. Cohen, S. Havlin, and H. E. Stanley. Optimal paths in disordered complex networks. *Phys. Rev. Lett.*, 91(16):168701, 2003.

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

[40] S. Butenko and W.E. Wilhelm. Clique-detection models in computational biochemistry and genomics. *European Journal of Operational Research*, 173:1–17, 2006.

[41] B. A. Carreras, V. E. Lynch, I. Dobson, and D. E. Newman. Critical points and transitions in an electric power transmission model for cascading failure blackouts. *Chaos*, 12(4):985–994, 2002.

[42] S. Chakrabarti, M. van den Berg, and B. Dom. Focused crawling: a new approach to topic-specific web resource discovery. *Computer Networks*, 31(11):1623–1640, 1999.

[43] Z. Y. Chen and X. F. Wang. Effects of network structure and routing strategy on network capacity. *Phys. Rev. E*, 73(3):036107, 2006.

[44] F. Chung and L. Lu. Connected components in random graphs with given degree sequences. *Annals of combinatorics*, 6:125–145, 2002.

[45] A. Clauset and C. Moore. How do networks become navigable? 2003. e-print cond-mat/0309415, http://xxx.lanl.gov/abs/cond-mat/0309415.

[46] V. Colizza, A. Barrat, M. Barthlemy, and A. Vespignani. The role of the airline transportation network in the prediction and predictability of global epidemics. *PNAS*, 103(7):2015–2020, 2006.

[47] N. Contractor, S. Wasserman, and K. Faust. Testing multi-theoretical multi-level hypotheses about organizational networks: An analytic framework and empirical example. *Academy of Management Review*, 31(3):681–703, 2006.

[48] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, 2^{nd} edition, 2001.

[49] L. F. Costa. Reinforcing the resilience of complex networks. *Phys. Rev. E*, 69(6):066127, 2004.

[50] P. Crucitti, V. Latora, and M. Marchiori. Model for cascading failures in complex networks. *Phys. Rev. E*, 69(4):045104, 2004.

[51] L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas. Comparing community structure identification. *Journal of Statistical Mechanics*, page P09008, 2005.

[52] M. Argollo de Menezes and A.-L. Barabási. Fluctuations in network dynamics. *Phys. Rev. Lett.*, 92(2):028701, 2004.

[53] O. Diekmann and J. Heesterbeek. *Mathematical Epidemiology of Infectious Diseases: Model Building, Analysis and Interpretation.* Wiley, New York, 2000.

[54] P. Dodds, R. Muhamad, and D. J. Watts. An experimental study of search in global social networks. *Science*, 301(5634):827–829, 2003.

[55] S. Dorogovtsev and J. F. F. Mendes. Scaling behaviour of developing and decaying networks. *Europhys. Lett.*, 52:33–39, 2000.

[56] S. N. Dorogovtsev and J. F. F. Mendes. Evolution of networks. *Adv. Phys.*, 51:1079–1187, 2002.

[57] S. N. Dorogovtsev and J. F. F. Mendes. Minimal models of weighted scale-free networks. *e-print cond-mat/0408343, http://lanl.arxiv.org/abs/cond-mat?papernum=0408343*, 2004.

[58] P. Echenique, J. Gomez-Gardenes, and Y. Moreno. Improved routing strategies for internet traffic delivery. *Phys. Rev. E*, 70(5):056105, 2004.

[59] P. Echenique, J. Gomez-Gardenes, and Y. Moreno. Dynamics of jamming transitions in complex networks. *Europhys. Lett.*, 71(2):325–331, 2005.

[60] P. Erdos and A.Renyi. On random graphs. *Publicationes Mathematicae*, 6:290–297, 1959.

[61] P. Erdos and A.Renyi. On the evolution of random graphs. *Magyar Tud. Mat. Kutato Int. Kozl.*, 5:17–61, 1960.

[62] P. Erdos and A.Renyi. On the strength of connectedness of a random graph. *Acta Math. Acad. Sci. Hungar.*, 12:261–267, 1961.

[63] R. Estrin, J. Govindan, Heidemann, and S. Kumar. Next century challenges: Scalable coordination in sensor networks.

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

[65] Y. Fan, M. Li, J. Chen, L. Gao, Z. Di, and J. Wu. Network of econophysicists:a weighted network to investigate the development of econophysics. *International Journal of Modern Physics B*, 18:2505–2511, 2004.

[66] G. Flake, S. Lawrence, and C. Lee Giles. Efficient identification of web communities. In *Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 150–160, 2000.

[67] O. Frank and D. Strauss. Markov graphs. *J. American Statistical Association*, 81:832–842, 1986.

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

[69] K. I. Goh, B. Kahng, and D. Kim. Universal behavior of load distribution in scale-free networks. *Phys. Rev. Lett.*, 87(27), 2001. 278701.

[70] K. I. Goh, J. D. Noh, B. Kahng, and D. Kim. Load distribution in weighted complex networks. *Phys. Rev. E*, 72(1):017102, 2005.

[71] R. Govindan and H. Tangmunarunkit. Heuristics for internet map discovery. *IEEE INFOCOM*, 3:1371–1380, 2000.

[72] M. Granovetter. The strength of weak ties. *American Journal of Sociology*, 78(6):1360–1380, 1973.

[73] R. Guimera and L. A. N. Amaral. Modeling the world-wide airport network. *Eur. Phys. J. B*, 38:381385, 2004.

[74] R. Guimera, A. Arenas, A. Díaz-Guilera, and F. Giralt. Dynamical properties of model communication networks. *Phys. Rev. E*, 66(2):026704, 2002.

[75] R. Guimera, A. Díaz-Guilera, F. Vega-Redondo, A. Cabrales, and A. Arenas. Optimal network topologies for local search with congestion. *Phys. Rev. Lett.*, 89(24):248701, 2002.

[76] R. Guimera, S. Mossa, A. Turtschi, and L. A. N. Amaral. The worldwide air transportation network: Anomalous centrality, community structure, and cities' global roles. *Proc. Nat. Acad. Sci.*, 102:7794–7799, 2005.

[77] A. Gulli and A. Signorini. The indexable web is more than 11.5 billion pages. In *WWW '05: Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 902–903. ACM Press, New York, USA, 2005.

[78] P. Hansen and B. Jaumard. Cluster analysis and mathematical programming. *Mathematical programming*, 79:191–215, 1997.

[79] J. Hasselberg, P. M. Pardalos, and G. Vairaktarakis. Test case generators and computational results for the maximum clique problem. *Journal of Global Optimization*, 3:463–482, 1993.

[80] B. Hendrickson and R. W. Leland. A multilevel algorithm for partitioning graphs. In *Supercomputing '95: Proceedings of the 1995 ACM/IEEE conference on Supercomputing*, page 28. ACM Press, New York, USA, 1995.

[81] P. W. Holland and S. Leinhardt. An exponential family of probability distributions for directed graphs. *J. American Statistical Association*, 76:33–65, 1981.

[82] P. Holme, M. Huss, and H. Jeong. Subnetwork hierarchies of biochemical pathways. *Bioinformatics*, 19:532–538, 2003.

[83] P. Holme and B. J. Kim. Attack vulnerability of complex networks. *Phys. Rev. E*, 65(5), 2002.

[84] R. Ferrer i Cancho, C. Janssen, and R. V. Solé. Topology of technology graphs: Small world patterns in electronic circuits. *Phys. Rev. E*, 64(4):046119, 2001.

[85] R. Ferrer i Cancho and R. V. Solé. *Statistical mechanics of complex networks*, chapter Optimization in complex networks, pages 114–126. Springer-Verlag, Berlin, 2003.

[86] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. *Proceedings of ACM MobiCom '00, Boston, MA*, pages 174–185, 2000.

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

[88] D. J. Johnson and M. A. Trick, editors. *Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge, Workshop, October 11-13, 1993*. American Mathematical Society, Boston, USA, 1996.

[89] Y.C. Lai K. Park and N. Ye. Characterization of weighted complex networks. *Phys. Rev. E*, 70(2):026109, 2004.

[90] V. Kalapala, V. Sanwalani, A. Clauset, and C. Moore. Scale invariance in road networks. *Phys. Rev. E*, 73:026130, 2006.

[91] G. Kan. *Peer-to-Peer Harnessing the Power of Disruptive Technologies*, chapter Gnutella. O'Reilly, Beijing, 2001.

[92] A.-M. Kermarrec, L. Massoulie, and A. J. Ganesh. Probabilistic reliable dissemination in large-scale systems. *IEEE Trans. on Parallel and Distributed Sys*, 14(3):248–258, 2003.

[93] B.W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *The Bell System Technical Journal*, 49:291–307, 1970.

[94] R. Kinney, P. Crucitti, R. Albert, and V. Latora. Modeling cascading failures in the north american power grid. *The European Physical Journal B*, 46:101–107, 2005.

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

[96] J. Kleinberg. The small-world phenomenon: An algorithmic perspective. *Proc. 32nd ACM Symposium on Theory of Computing*, pages 163–170, 2000.

[97] J. Kleinberg. Small-world phenomena and the dynamics of information. *Advances in Neural Information Processing Systems*, 14:431–438, 2001.

[98] J. Kleinberg. Complex networks and decentralized search algorithms. *Proceedings of the International Congress of Mathematicians*, 3:1019–1044, 2006.

[99] D. Koschützki, K. A. Lehmann, L. Peeters, S. Richter, D. Tenfelde-Podehl, and O. Zlotowski. *Network Analysis*, chapter Centrality Indices, pages 16–61. Springer-Verlag, Berlin, 2005.

[100] A. E. Krause, K. A. Frank, D. M. Mason, R. E. Ulanowicz, and W. W. Taylor. Compartments revealed in food-web structure. *Nature*, 426:282–285, 2003.

[101] J. Kulik, W. R. Heinzelman, and H. Balakrishnan. Negotiation-based protocols for disseminating in wireless sensor networks. *Wireless Networks*, 8:169–185, 2002.

[102] R. Kumar, P. Raghavan, S. Rajalopagan, D. Sivakumar, A. Tomkins, and E. Upfal. The web as a graph. *Proceedings of the nineteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 1–10, 2000.

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

[104] J. Leskovec, L. A. Adamic, and B. A. Huberman. The dynamics of viral marketing, 2005. e-print physics/0509039, http://lanl.arxiv.org/abs/physics?papernum=0509039.

[105] C. Li and G. Chen. Network connection strengths: Another power-law? 2003. e-print cond-mat/0311333, http://lanl.arxiv.org/abs/cond-mat?papernum=0311333.

[106] W. Li and X. Cai. Statistical analysis of airport network of china. *Phys. Rev. E*, 69(4):046106, 2004.

[107] D. Liben-Nowell, J. Novak, R. Kumar, P. Raghavan, and A. Tomkins. Geographic routing in social networks. *Proc. Natl. Acad. Sci.*, 102(33):11623–11628, 2005.

[108] A. L. Lloyd and R. M. May. How viruses spread among computers and people. *Science*, 292:1316–1317, 2001.

[109] F. Menczer. Growing and navigating the small world web by local content. *Proc. Natl. Acad. Sci.*, 99(22):14014–14019, 2002.

[110] V. P. Mhatre, C. Rosenberg, D. Kofman, R. Mazumdar, and N. Shroff. A minimum cost heterogeneous sensor network with a lifetime constraint. *IEEE Transactions on Mobile Computing*, 4(1):4–15, 2005.

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

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

[113] Y. Moreno, J. B. Gomez, and A. F. Pacheco. Instability of scale-free networks under node-breaking avalanches. *Europhys. Lett.*, 58(4):630–636, 2002.

[114] Y. Moreno, R. Pastor-Satorras, A. Vazquez, and A. Vespignani. Critical load and congestion instabilities in scale-free networks. *Europhys. Lett.*, 62(2):292–298, 2003.

[115] A. E. Motter. Cascade control and defense in complex networks. *Phys. Rev. Lett.*, 93(9):098701, 2004.

[116] A. E. Motter and Y. Lai. Cascade-based attacks on complex networks. *Phys. Rev. E*, 66(6):065102, 2002.

[117] M. E. J. Newman. *Handbook of Graphs and Networks*, chapter Random graphs as models of networks.

[118] M. E. J. Newman. Models of small world. *Journal Statistical Physics*, 101:819–841, 2000.

[119] M. E. J. Newman. Scientific collaboration networks: I. network construction and fundamental results. *Phys. Rev. E*, 64(1):016131, 2001.

[120] M. E. J. Newman. Scientific collaboration networks: Ii. shortest paths, weighted networks, and centrality. *Phys. Rev. E*, 64(1):016132, 2001.

[121] M. E. J. Newman. Assortative mixing in networks. *Phys. Rev. Lett.*, 89:208701, 2002.

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

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

[124] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69(2):026113, 2004.

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

[126] J. D. Noh and H. Rieger. Stability of shortest paths in complex networks with random edge weights. *Phys. Rev. E*, 66(6):066127, 2002.

[127] T. Ohira and R. Sawatari. Phase transition in a computer network traffic model. *Phys. Rev. E*, 58(1):193–195, 1998.

[128] Committee on network science for future army applications. *Network Science*. The National Academies Press, 2005.

[129] E. Almaas P. J. Macdonald and A. L. Barabási. Minimum spanning trees on weighted scale-free networks. 2004. e-print cond-mat/0405688, http://lanl.arxiv.org/abs/cond-mat?papernum=0405688.

[130] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435:814–818, 2005.

[131] P. M. Pardalos and J. Xue. The maximum clique problem. *Journal of Global Optimization*, 4:301–328, 1994.

[132] R. Pastor-Satorras and A. Vespignani. Epidemic dynamics and endemic states in complex networks. *Phys. Rev. E*, 63(6):066117, 2001.

[133] R. Pastor-Satorras and A. Vespignani. Epidemic spreading in scale-free networks. *Phys. Rev. Lett.*, 86:3200–3203, 2001.

[134] R. Pastor-Satorras and A. Vespignani. Epidemic dynamics in finite size scale-free networks. *Phys. Rev. E*, 65(3):035108, 2002.

[135] R. Pastor-Satorras and A. Vespignani. Immunization of complex networks. *Phys. Rev. E*, 65(3):036104, 2002.

[136] R. Pastor-Satorras and A. Vespignani. *Handbook of Graphs and Networks*, chapter Epidemics and immunization in scale-free networks. Wiley-VCH, Berlin, 2003.

[137] R. Pastor-Satorras and A. Vespignani. *Evolution and structure of the Internet: A statistical physics approach*. Cambridge University Press, 2004.

[138] G. Paul, T. Tanizawa, S. Havlin, and H. E. Stanley. Optimization of robustness of complex networks. *Eur. Phys. Journal B*, 38:187–191, 2004.

[139] S.L. Pimm. *Food Webs*. The University of Chicago Press, 2 edition, 2002.

[140] A. Pothen, H. Simon, and K. Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM J. Matrix Anal.*, 11(3):430–452, 1990.

[141] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi. Defining and identifying communities in networks. *Proc. Natl. Acad. Sci.*, 101:2658–2663, 2004.

[142] U. N. Raghavan and S. R. T. Kumara. Decentralized topology control algorithms for connectivity of distributed wireless sensor networks. *International Journal of Sensor Networks*, 2:201–210, 2007.

[143] E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, and A. L. Barabási. Hierarchical organization of modularity in metabolic networks. *Science*, 297:1551–1555, 2002.

[144] A. R. Ravindran, editor. *Operations Research and Management Science Handbook*, chapter Complexity and Large-scale Networks. CRC press. in print.

[145] M. Ripeanu, I. Foster, and A. Iamnitchi. Mapping the gnutella network: Properties of large-scale peer-to-peer systems and implications for system design. *IEEE Internet Computing Journal*, 6:50–57, 2002.

[146] A. W. Rives and T. Galitskidagger. Modular organization of cellular networks. *Proc. Natl. Acad. Sci.*, 100(3):1128–1133, 2003.

[147] M. Rosvall, A. Trusina, P. Minnhagen, and K. Sneppen. Networks and cities: An information perspective. *Phys. Rev. Lett.*, 94:028701, 2005.

[148] M. L. Sachtjen, B. A. Carreras, and V. E. Lynch. Disturbances in a power transmission system. *Phys. Rev. E*, 61(5):4877–4882, 2000.

[149] O. Sandberg. Distributed routing in small-world networks. *Proceedings of the 8th Workshop on Algorithm Engineering and Experiments (ALENEX)*, pages 144–155, 2006.

[150] B. Shargel, H. Sayama, I. R. Epstein, and Y. Bar-Yam. Optimization of robustness and connectivity in complex networks. *Phys. Rev. Lett.*, 90(6):068701, 2003.

[151] Ö. Simsek and D. Jensen. Decentralized search in networks using homophily and degree disparity. *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, pages 304–310, 2005.

[152] B. K. Singh and N. Gupte. Congestion and decongestion in a communication network. *Phys. Rev. E*, 71(5):055103, 2005.

[153] T. A. B. Snijders. Markov chain monte carlo estimation of exponential random graph models. *J. Social Structure*, 3(2):1–40, 2002.

[154] S. Sreenivasan, R. Cohen, E. Lopez, Z. Toroczkai, and H. E. Stanley. Communication bottlenecks in scale-free networks, 2006. e-print cs.NI/0604023, http://xxx.lanl.gov/abs/cs?papernum=0604023.

[155] D. Strauss. On a general class of models for interaction. *SIAM Review*, 28:513–527, 1986.

[156] H. P. Thadakamalla, R. Albert, and S. R. T. Kumara. Search in weighted complex networks. *Phys. Rev. E*, 72(6):066128, 2005.

[157] H. P. Thadakamalla, R. Albert, and S. R. T. Kumara. Search in spatial scale-free networks. *New Journal of Physics*, 9:190, 2007.

[158] H. P. Thadakamalla, U. N. Raghavan, S. R. T. Kumara, and R. Albert. Survivability of multi-agent based supply networks: A topological perspective. *IEEE Intelligent Systems*, 19:24–31, 2004.

[159] Z. Toroczkai and K. E. Bassler. Network dynamics: Jamming is limited in scale-free systems. *Nature*, 428:716, 2004.

[160] A. X. C. N. Valente, A. Sarkar, and H. A. Stone. Two-peak and three-peak optimal complex networks. *Phys. Rev. Lett.*, 92(11):118702, 2004.

[161] V. Venkatasubramanian, S. Katare, P. R. Patkar, and F. Mu. Spontaneous emergence of complex optimal networks through evolutionary adaptation. *Computers & Chemical Engineering*, 28(9):1789–1798, 2004.

[162] W. Vogels, R. van Renesse, and K. Birman. The power of epidemics: robust communication for large-scale distributed systems. *SIGCOMM Comput. Commun. Rev.*, 33(1):131–135, 2003.

[163] X. F. Wang and J. Xu. Cascading failures in coupled map lattices. *Phys. Rev. E*, 70(5):056113, 2004.

[164] S. Wasserman and K. Faust. *Social Network Analysis*. Cambridge University Press, 1994.

[165] S. Wasserman and P. Pattison. Logit models and logistic regressions for social networks 1: An introduction to markov random graphs and p*. *Psychometrika*, 61:401–426, 1996.

[166] D. J. Watts. A simple model of global cascades on random networks. *Proc. Natl. Acad. Sci.*, 99(9):5766–5771, 2002.

[167] D. J. Watts. *Six degrees: The science of a connected age*. W. W. Norton & Company, 2003.

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

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

[170] H. S. Wilf. *Generating Functionology*. Academic, Boston, 1990.

[171] D. M. Wilkinson and B. A. Huberman. A method for finding communities of related genes. *Proc. Natl. Acad. Sci.*, 101:5241–5248, 2004.

[172] M. Yarvis, N. Kushalnagar, H. Singh, A. Rangarajan, Y. Liu, and S. Singh. Exploiting heterogeneity in sensor networks. *Proceedings of 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, 2:878–890, 2005.

[173] S. H. Yook, H. Jeong, and A. L. Barabási. Modelling the internet's large-scale topology. *Proc. Nat. Acad. Sci.*, 99(21):13382–13386, 2002.

[174] S. H. Yook, H. Jeong, A. L. Barabási, and Y. Tu. Weighted evolving networks. *Phys. Rev. Lett.*, 2001:5835–5838, 86.

[175] H. Zhang, A. Goel, and R. Govindan. Using the small-world model to improve freenet performance. *Computer Networks*, 46(4):555–574, 2004.

Vita

Hari Prasad Thadakamalla

Education

The Pennsylvania State University, University Park, PA. (Aug. 2001 - Dec. 2007)
Ph.D., Industrial Engineering and Operations Research, *December* 2007 (expected).
M.A., Mathematics and minor in Applied Mathematics, *December* 2007 (expected).
M.S., Industrial Engineering, August 2004.

Indian Institute of Technology (I.I.T.), Madras, India. (Jul. 1997 - May 2001)
Bachelor of Technology, Mechanical Engineering, May 2001.

Academic distinctions and Honors

- Received NRW Undergraduate Science Award 2004 given by University of Dortmund, Germany in conjunction with Ministry of Science and Research, Düsseldorf, Germany
- Awarded full scholarship (Merit and Means) by Indian Institute of Technology, Madras, during my junior and senior year
- Selected for the student honorarium and placed 3rd in the posters category at the International Workshop and Conference on Network Science, 2006.

Selected Publications

- Thadakamalla, H. P., Kumara S. R. T., and Albert, R., Complexity and large-scale networks, Ravindran, A. R. Ed., *Operations Research and Management Science Handbook*, CRC press (in print).
- Thadakamalla, H. P., Albert, R., and Kumara S. R. T., Search in spatial scale-free networks, *New Journal of Physics* **9**, p. 190, 2007.
- Thadakamalla, H. P., Albert, R., and Kumara S. R. T., Search in weighted complex networks, *Phys. Rev. E.* **72**, p. 066128, 2005.
- Thadakamalla, H. P., Raghavan, U. N., Kumara, S. R. T., and Albert, R., Survivability of multi-agent based supply networks: A topological perspective, *IEEE Intelligent Systems* **19**(5), p. 24, 2004.