2012

Network Survivability Analysis: Coarse-Graining And Graph-Theoretic Strategies

Teshome Feyessa
North Carolina Agricultural and Technical State University

NETWORK SURVIVABILITY ANALYSIS: COARSE-GRAINING AND GRAPH-THEORETIC STRATEGIES

by

Teshome Feyessa

A dissertation submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY

Department: Electrical and Computer Engineering
Major: Electrical Engineering
Major Professor: Dr. Marwan Bikdash

North Carolina A&T State University
Greensboro, North Carolina
2012

ABSTRACT

Feyessa, Teshome. NETWORK SURVIVABILITY ANALYSIS: COARSE-GRAINING AND GRAPH-THEORETIC STRATEGIES. **(Major Advisor: Marwan Bikdash)**, North Carolina Agricultural and Technical State University

Survivability is the ability of a system to continuously deliver essential services despite attacks, failures, or accidents that damage a significant portion of the system. Most of existing survivability measures evaluate global impact of faults on a network by averaging the loss in performance over the whole network. These approaches undermine the fact that a fault is not expected to impact all parts of the network equally. Impacts of some faults are contained in local neighborhoods while others have a global reach. In addition, graph algorithms are often computationally intensive, and even polynomial time algorithms get impractical for a fair sized network.

In this dissertation, the interplay between geographic information about the network and the principal properties and structure of the underlying graph are used to quantify the structural and functional survivability of the network. This work focuses on the local aspect of survivability by studying the propagation of loss in the network as a function of the distance of the fault from a given origin-destination node pair.

Geographic-based partitioning and graph-based representations of the interactions of the partitions are used to build a coarsened network. The partitions are designed to behave as a subnetwork. A complexity reduction in network computation is achieved by performing the desired computation on the subnetworks and the coarsened network. The overall network parameters are determined by merging the probability distributions in the subnetworks with the parameters of the coarsened network.

School of Graduate Studies
North Carolina Agricultural and Technical State University

This is to certify that the Doctoral Dissertation of

Teshome Feyessa

has met the dissertation requirements of
North Carolina Agricultural and Technical State University

Greensboro, North Carolina
2012

Approved by:

Dr. Marwan Bikdash Major Professor	Dr. Gary Lebby Committee Member
Dr. Numan Dogan Committee Member	Dr. John Kelly Committee Member
Dr. Robert Li Committee Member	Dr. John Kelly Department Chairperson

Dr. Sanjiv Sarin
Associate Vice Chancellor of Research
and Graduate Dean

BIOGRAPHICAL SKETCH

Teshome Feyessa was born on October 25, 1983 in Bahir Dar, Ethiopia. He received his Bachelor of Science degree in Electrical Engineering from Bahir Dar University, Faculty of Engineering, in 2006. He received the Master of Science degree from the Department of Electrical and Computer Engineering at the North Carolina Agricultural and Technical State University in 2009. He is a candidate for a Ph.D. in the Department of Electrical and Computer Engineering at the North Carolina Agricultural and Technical State University.

ACKNOWLEDGEMENTS

My heartfelt gratitude goes to my family for their love and support. I am indebted to my advisor, Dr. Bikdash, for his guidance in the entirety of this work in addition to his invaluable technical assistance. My sincere appreciation also goes to Dr. Lebby, Dr. Dogan, Dr. Kelly and Dr. Li for their advice and serving on my dissertation committee.

This work is supported by Pennsylvania State University and The Defense Threat Reduction Agency under contract DTRA01-03-D-0010/0020 and Subcontract S03-34. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the DTRA or the U.S. Government.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURES **PAGE**

LIST OF TABLES

TABLES **PAGE**

CHAPTER 1

INTRODUCTION

Large-scale highly-distributed and unbounded network environments, like the internet and transportation networks present opportunities for massive propagation of failures as well as the opportunity to recover from failures due to high connectivity. This raises the need to incorporate survivability capabilities to the system organization. Survivability is the ability of a system to continuously deliver essential services and maintain essential properties such as integrity, confidentiality and performance despite attacks, failures or accidents that damage or compromise a significant portion of the system [1]. Attacks can be physical, such as the destruction of nodes and/or links, but can also include intrusions, probes and denial of service. Failures, which are mostly internally generated, include system deficiencies, software design errors, hardware degradation, human error and corrupted data. Accidents are usually random and external, such as natural disasters [1, 2].

A network can be defined as an object composed of elements and interactions or connections between those elements. Network theory is used to model objects (nodes) and maps their relationship (links). It enables better understanding of behavior of the network that would have otherwise been impossible to predict from looking at individual parts. Networks are found everywhere, such as telecommunication networks, social networks, neural networks, transportation networks, power networks, distribution networks and financial network. A node can represent an intersection of roads, a telecommunication switch, a computer or a person. Similarly, a link can represent a road, a fiber optic cable, an Ethernet cable and friendship

Network theory has its root in graph theory and linear algebra. A graph, $G(V, E)$, made up of a node set V, and a link set E, is a natural means to model networks mathematically. Linear algebra is used in the matrix representation and manipulation of networks. The main characteristics of a survivable network include resistance to attacks, robustness under attacks, attribute balancing and redundancy. There is also an emergent behavior requirement on survivable networks.

A key issue in survivability analysis is the quantification and measurement of survivability. In an early work [3] on the survivability of command, control and communication (C^3) networks, survivability was described in terms of the existence of communication paths, the number of nodes in the largest connected section, the shortest surviving paths, the fraction of nodes that can still communicate, and the maximum time required to transmit messages after attack. In Markovian models, network survivability is quantified by combining various performance models of the different failure propagation and recovery phases [4].

Most of the survivability measures developed so far use the average or expected values of quantities computed over the entire network, thus giving a global view of the network survivability. But the extent of the impact of a fault on its neighborhood and the reach of the impact remain uncharacterized. This work proposes a way of filling this gap with new measures of local as well as global survivability of a network that can characterize the effect of a fault at different scales of distance. The proposed survivability measures use both topology-based and traffic-based performance measures.

Large and complex networks are computationally expensive to visualize, and pre-

dicting their response to faults and attacks is even more challenging. In addition, in infrastructure networks interactions between different sections of the network are at times more important than node-to-node interactions. For instance, the flow between two road intersections is much less important than the flow between two parts of a city or even two different cities.

There is work on graph partitioning and network coarsening that tries to address these problems. The focus of most of network coarsening approaches is mainly to maintain much of the properties of the original network, such as degree distribution and topology [5], rather than maintaining the geographic information. Although, there are geographic-based coarse-graining approaches [6], they are more reliant on the geometry of the network than on the actual geographic locations. This work presents a combination of graph partitioning, in the form of node clustering, and coarsening procedures to reduce the size of a network. The network's nodes are clustered according to their geographic locations and their connectivity constraint. The geographic coarse-graining approach is used to estimate network level measures and implement a shortest-path hierarchical routing.

This dissertation addresses three main challenges of network survivability analysis. The first key issue is the formulation of graph-theoretic survivability measures and analyzing the propagation of performance loss in a network. A wide range of options are considered to quantify network survivability. This work focuses on the local aspects of survivability, in addition to well known global survivability measures. The second challenge is to reduce the computational complexity of survivability analysis that arise due to the size of the network and the number of fault scenarios. A geographic-based coarsening approach

is implemented to address this challenge on different levels of network resolution. Coarsening is also used to obtain a reduced-order model of the network. The third challenge is the combinatorial nature of fault scenarios. Most fault analysis in networks are limited to random single failures. This work proposes ways of creating attack scenario and simulation of multiple random and targeted as well as geographically localized failures.

As an example of real-world spatial networks, the highway networks of some of the South East states of the US are considered. The basic network model of the highway system represents any kind of intersection between two roads as a node and the actual roads as links. GIS data of the national highway planning network (NHPN) is obtained from U.S. Department of Transportation, Federal Highway Administration website[1].

1.1 Synopsis

This dissertation is organized as follows. Chapter 2 presents basic concepts in network theory with emphasis on network models and network essential problems that are useful in characterizing the survivability of a network. In Chapter 3, global measures of network robustness based on the graph topology are developed. The response of a network under different types of perturbation as a function of these measures is also part of this chapter. In Chapter 4, a local survivability measure is formulated to quantify the containment and propagation of an impact of a fault. In Chapter 5, a network coarsening theory is presented and geographic-based partitioning is employed to obtain a reduced-order model of a network. Chapter 6 presents the emergence of small-world property in road network at higher levels of coarsening. Finally, the conclusions are presented in Chapter 7.

[1] See http://www.fhwa.dot.gov/planning/nhpn/

CHAPTER 2

BACKGROUND

2.1 Graph Basics

Graphs are made up of set of vertices (usually called nodes in networking) and the set of edges (links) connecting them. The most common notation for graphs is $G(V, E)$, where $V(G)$ is the vertex set of graph G with cardinality n and $E(G)$ is the edge set of graph G with cardinality m. The order of a graph $n(G)$ is the number of vertices in G. A graph's nodes and links represent different concepts based on the network it represents. For instance, in transportation network roads and highways are represented by edges while intersections are nodes; in telecommunication network nodes can be computers, routers, repeaters, etc. and edges can be cables, fiber optics, wireless media, etc.; in social networks nodes can be people and groups whereas links can be relationship between people such as colleague, friend, family, etc.

Different prefixes to the word graph are added to show a special type of graph, such as multigraph. A Multigraph is a graph containing parallel edges, i.e., there is a repetition in the edge set $E(G)$. In a weighted graph the edge set is mapped to a weighing function. Undirected graphs have edges that do not have any specific direction of connection, as opposed to directed graph. Subgraphs are graphs made out of graphs. Mathematically a subgraph $G'(V', E')$ is defined as a graph whose vertex set $V'(G')$ and edge set $E'(G')$ are subsets of $V(G)$ and $E(G)$ respectively. If $E'(G')$ contains all of the edges in $E(G)$ joining all the vertices in $V'(G')$ it is called an induced subgraph. There are also planar graphs, non planar graphs, Bipartite graphs, cliques, etc.

The *Degree* of a vertex i, δ_i, is the number of edges in E that have i as a source or end-vertex. In directed graphs, the in-degree of a node is not necessarily equal to its out-degree. If $\delta_i = k$ for all $i \in V$, where k is a non-negative integer constant, the corresponding graph is called a *regular graph*.

A *walk* is an alternating sequence of vertices and edges starting from one vertex and ending on another vertex. If the start vertex and end vertex are same the walk is called a cycle. A walk that does not have any repeated edge is called a *path*. Two paths are said to be internally disjoint if they do not share a non endpoint vertex, for instance, the dotted paths in Figure 2.1d. A graph is said to be strongly *connected* if there exists a path between every pair of vertices. The graphs in Figures 2.1c and d are strongly connected. A weakly connected graph is a digraph that is not connected but its equivalent undirected graph is strongly connected.

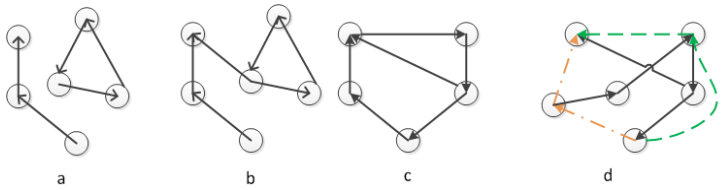

Figure 2.1. Graph connectivity. (a) Unconnected (b) Weakly connected (c) Strongly connected (d) Internally disjoint paths

A *flow network* is a directed graph with a function assigning non-negative capacities to the edges, and two distinct vertices source and sink. A separating set or vertex cut of a graph G is a set S subset of $V(G)$ such that $G - S$ has more than one component. A graph G is k-connected if every vertex cut has at least k vertices. Connectivity of G, $k(G)$, is the maximum k such that G is k-connected. A clique has no separating set. For bipartite graphs

$k(K_{m,n}) = min\{m, n\}$. A disconnecting set of edges is a set F subset of $E(G)$ such that $G - F$ has more than one component. An edge cut $[S, S']$ is a set of edges whose removal separates the vertex set into S and $S' = V - S$ where S is a non-empty proper subset of $V(G)$. A graph is k-edge-connected if every disconnecting set has at least k edges. The edge connectivity of G, $k'(G)$, is the minimum size of a disconnecting set or the maximum k such that G is k-edge connected. A block of a graph G is a maximally connected subgraph of G that has no cut-vertex. If G itself is connected and has no cut-vertex, then G is a block. Cut-vertex and cut-edge are vertex cuts and edge cuts of size one respectively, that increase the number of components of the graph.

Theorem 1 Let, $k(G)$, $k'(G)$, and $\delta(G)$ be the vertex connectivity, the edge connectivity and the minimum degree of a graph G respectively. Then $k(G) \leq k\prime(G) \leq \delta(G)$.

In computer simulation and analysis a graph is typically represented by its adjacency-matrix, adjacency list and node-arc incidence matrix. The adjacency matrix, A, of a graph is an n by n matrix, such that $A(i, j)$ is the number of links that are incident on both node i and node j.

2.2 Network Essential problems

2.2.1 Connected components

In a network based systems, there are essential problems that need to be addressed to have a full understanding of the system and to design a stable system. The first of such problems is finding connected components of a network. A connected component of a graph is an induced subgraph G' that is connected and maximal, i.e., there exists no connected subgraph G'' with vertex set V'' that is superset of V'. The problem of connectivity is used to analyze graph connectivity at the face of edge(s) and/or node(s) failure with min-

imum edges when connections are expensive. Loops are irrelevant for connection. The problem of connectivity is deeply rooted in cut and cutset formulation. Tarjan's algorithm uses depth first search to find the strongly connected components of a network [7].

2.2.2 Shortest Paths

In weighted directed graphs, a distance between two vertices is measured by the sum of the weights of the edges (and vertices if they are also weighted) that are in the path. The shortest path problem is defined as a problem of finding a path that has the smallest weight. It can be defined between a single source and a single destination, or a single source and the rest of the vertices, or between all vertices against every other vertex.

In unweighted graphs, the single-source shortest paths can be obtained using a breadth first search. If there are no negative weights in the graph Dijkstra's algorithm is the most common solution to the single source shortest path problem. If the graph contains edges with negative weights but no negative cycles a modified Dijkstra or a Bellman-Ford algorithm can be used to find shortest paths. All of these algorithms share the relaxation technique as a mode of discovering better paths [8–10].

The solution to the all-pair shortest path problem can involve running single source shortest paths for all sources or using dynamic programming like the Floyd-Warshall algorithm or Johnson's algorithm in case of sparse graphs [10]. The shortest-path algorithms listed here are developed with static weights in mind. Time-dependent shortest-path algorithms have also been researched, especially in transportation networks [11]. The average path and the longest of the shortest paths over all node-pairs in a network are called the network's characteristic path length and diameter respectively.

2.2.3 Shortest Disjoint Paths

Shortest path problems have various applications ranging from physical computation of shortest distance transportation paths to minimum delay paths in telecommunication networks. But finding the shortest path does not suffice to quantify survivability. To ensure survivability maximally disjoint paths are important in the network, hence computation of shortest pair or, more generally, set of shortest disjoint paths is essential. This problem has a wide range of application. For instance, dispatching hazmat trucks requires minimizing the interaction risks and delays associated with potential accidents while minimizing the total transit costs. Similarly, in a congested and unreliable communication network, duplicate routing of packets via disjoint paths is necessary. A network is called k-node survivable if at least k node-disjoint paths between all pairs of nodes exist [12].

The shortest-pair disjoint paths problem is defined as finding a pair of paths between two vertices that are independent of each other and are "shortest". If the path independence is based on edge disjointedness, the paths are called shortest pair of edge-disjoint paths. Node-disjoint paths, also called internally disjoint paths, do not have any common node except the start and end nodes [9]. From an optimization point of view, the shortest k-disjoint paths can be seen as a minimum sum (min-sum) problem, that minimize the total cost of the paths, or a minimum maximum (min-max) problem, that minimize the longest path [13]. The prominent work in obtaining minimum cost pair of edge-disjoint paths is done by Suurballe using multiple invocation of the Dijkstra algorithm and graph transformation of a modified graph [14]. Bhandari has provided an improved algorithm that does not require graph transformation by using a modified Dijkstra or breadth first search algo-

rithm to obtain the paths [9].

2.2.4 Network Cut and Flow

A flow network is a directed graph with a function assigning non-negative capacities to the edges. A flow is computed between two distinct vertices, called source and sink, that satisfy capacity constraints on the edges. The main objective of a network flow problem is finding a feasible flow pattern that maximizes the source to destination total flow. The prominent work of Ford and Fulkerson [15], Kirchoff's flow conservation law, and linear programming are widely used to address network flow problems. Network flow problems are found from classical problems such as operations research, transportation, water pipeline, communication flow to more recent problems such as sky survey, binary image reconstruction, radiation therapy and many more.

In general, a flow unit that is relevant for one origin-destination pair cannot be substituted for another unit corresponding to another pair. Therefore a multi-commodity flow must be considered. Consider a set of K commodities, $J_k = (i_k, j_k, \delta_k, f_k^*)$, where i_k, j_k, δ_k, f_k^* are the source node, destination node, demand for J_k and optimal flow of J_k respectively. The optimal flow over the network is constrained by

$$f_k \geq \delta_k \tag{1}$$

$$\sum_{j' \in V} f_k(i', j') - \sum_{j' \in V} f_k(j', i') = \begin{cases} f_k^*, & i' = i_k \\ 0 & \text{otherwise} \\ -f_k^*, & i' = j_k \end{cases} \tag{2}$$

$$c(i', j') \geq \sum_{k=1}^{K} f_k(i', j') \geq 0, \ (i', j') \in E, \tag{3}$$

where $c(i', j')$ and $f_k(i', j')$ are the capacity of link (i', j') and the flow of commodity J_k over it respectively. Equations (2) and (3) represent the conservation of flow and the flow

capacity constraints.

The objective in multi-commodity flow problems is often cost minimization, or flow maximization when there are no demands specified. This is an NP-complete problem for more than one commodity. Therefore, for the sake of simplicity, and because most of the conclusions are expected to extend to the multi-commodity flow, the maximum feasible single-commodity flow (max-flow) is the most common flow problem considered. Hence, the graph can be rewritten as $G(V, E, c, f)$, where c and f are the capacity and flow functions respectively. The set of the flows $\{f_{ij}\}$ is called the flow pattern. A feasible flow in a network is a flow pattern that satisfy the constraint and conservation equations, i.e., the difference of flow coming and going out of every node except source and sink is zero and flow in every arc is less than the capacity of the arc [16, 17].

The maximum network flow problem for a source s and sink t is defined as finding a flow pattern with an objective of maximizing the flow from the source to destination, f_{st}, while satisfying the conservation equations and the link capacity constraints. The above problem can be formulated as a linear program.

$$\max f_{st}, \text{ subject to} \tag{4}$$

$$\sum_{j \in V} f(i,j) - \sum_{j \in V} f(j,i) = \begin{cases} f_{st}, & i = s \\ 0 & \text{otherwise} \\ -f_{st}, & i = t \end{cases} \tag{5}$$

$$c(i,j) \geq f(i,j) \geq 0, \ (i,j) \in E. \tag{6}$$

By definition, $[S, T]$ specifies the set of edges having one endpoint in S and the other in T. Such a cut is called $s - t$ cut, where $s \in S$, $t \in T$, and $T = V - S$. For two distinguished nodes s and t in a directed graph, an $s - t$ cutset is a minimal set of edges whose removal breaks all the path from s to t. An $s - t$ cut is specified with its capacity.

Capacity of a cut in a network described by $G(V, E, c, f)$ is the total flow capacity of set $[S, T]$. Hence, the minimum $s-t$ cut is defined as a cut having the least capacity among all possible $s-t$ cuts. Ford and Fulkerson's theorem [18] states that the maximum flow from node s to node t, in a capacitated network is equal to the capacity of the minimum $s-t$ cut.

The classic solutions for max-flow min-cut problem use the Ford-Fulkerson labeling algorithm to systematically search flow augmenting paths and increase the flow in these paths [17, 18]. A flow augmenting path is a path all of whose forward edges are unsaturated, i.e., $f(i,j) < c(i,j)$, and the reverse edges are not flowless. A flow is said to be maximum if there exists no augmenting path with respect to it.

2.2.5 Centrality

Centrality indices are quantifications of the fact that some nodes/edges are more central or more important in a network than others. Different centrality indices are suitable for different applications, but most of them have structural significance and require that the network be connected [19]. Centrality indices that involve volume or length of a walk are usually referred to as radial. Examples include degree-like and closeness-like measures. Indices based on the number of paths passing through a node, such as the betweenness measure, are called medial [20].

A large amount of work in centrality comes from the social network studies [21–23]. Centrality in such networks is usually interpreted as a measure of power and social stratification. There are also many instances of centrality applications in biological networks [24, 25], communication networks [26, 27], power networks [28] and transportation net-

works [29, 30]. In spatial networks, centrality measures are used in developing design requirement and studying vulnerability. In vulnerability analysis, centrality is used to identify critical locations and "vulnerability backbones" in the network and to assess how well the network is distributed. Removing central nodes generally leads to an increase in diameter, a reduction of flow and a decrease in structural connectivity [31].

The simplest centrality measure is node degree, that is, the number of neighbors a node has. Therefore the degree-centrality index is the degree vector of the graph. In contrast, spectral centrality measures, e.g., Bonacich centrality and α-centrality, take the importance of these neighbors into consideration [22]. Eigenvectors are important in spectral analysis of network and centrality measures [32]. The eigenvector centrality of a node i, c_i^E, can be defined [22] as a quantity satisfying

$$c_i^E = \frac{1}{\lambda} \sum_{j=1}^{n} A_{ij} c_j^E \tag{7}$$

where λ is a constant (an eigenvalue), n is the number of nodes in the network and A is the adjacency matrix.

Equation (7) can be rewritten as $\lambda C^E = AC^E$, where C^E a is vector of centrality indices. Both λ and elements of C^E must be nonnegative. The Perron-Frobenius theorem and its extensions [33] state that, a symmetric irreducible matrix with nonnegative entries has a simple real maximum eigenvalue $\lambda_{\max}$ and the entries of the corresponding eigenvector $\upsilon_{\max}$ are all positive. This result is directly applicable to the adjacency matrix of a connected graph. Similar, but weaker, results are available for digraphs. A matrix is irreducible if it could not be made block diagonal by row and column permutations. In short, $\upsilon_{\max}$ can be interpreted as a centrality vector.

Closeness-based centrality finds the distance center or the median of a graph. It has application in facility location [21], package delivery [23] and similar operations research problems. It is computed by summing up the distances from the candidate node to all remaining nodes. Let d_{ij} be length of shortest path between nodes i and j, then closeness centrality c_i^C is

$$c_i^C = \sum_{j=1}^{n} d_{ij} \tag{8}$$

The lesser the sum is, the more central is the node.

Betweenness is one of the most prominent centrality measures. It measures the influence/brokerage of a node over the connection of other nodes by summing up the fraction of shortest paths between the other nodes that pass through it [34,35]. This definition qualifies betweenness as a medial measure [20]. Given any two nodes i and j, one can compute the number of shortest paths, P_{ij}, between i and j that satisfy a given criterion such as a threshold on path length. Out of all the possible shortest paths, some will pass through the node h whose centrality is considered. Let P_{ij}^h denote the number of these paths, then the shortest path betweenness centrality of node h is defined as

$$c_h^B = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{P_{ij}^h}{P_{ij}} \text{ for } i, j \neq h \tag{9}$$

There are also many more centrality measures but they are usually variants of the indices defined above. Variants of shortest path betweenness can be obtained by including influence of endpoints, sources and targets, bounding or scaling the shortest distance, considering edge betweenness and group betweenness [34]. Sometimes betweenness is defined as a communicability betweenness. Communicability betweenness takes all paths into account rather than just the shortest paths [35].

Delta centrality, proposed in [36], is defined as the decrease in generic cohesiveness measure of the graph when the node is deactivated or removed from the network.

2.2.6 Graph Clustering

The average clustering coefficient of a network, which is the arithmetic mean of the clustering coefficient of the nodes, measures the tendency of the network to form tightly connected neighborhoods. The clustering coefficient of a node is defined as the ratio of number of triangles of which the node is a member of to the number of triangles the node could possibly participate in [37, 38]. Thus, the clustering coefficient of node i is

$$C_i = \frac{\sum_{j \neq i} \sum_{h \neq i,j} A_{ij} A_{ih} A_{hj}}{\delta_i(\delta_i - 1)} \tag{10}$$

2.2.7 Graph Spectra

Eigenvalues of matrices in a graph, especially the adjacency matrix, the Laplacian matrix and the normalized Laplacian matrix reflect structural properties about the graph. For instance, adjacency matrix is useful for counting paths of certain length in a graph, number of spanning trees and connected components can be determined from the Laplacian, and the normalized Laplacian enables recognition of connected components and bipartite structures [43]. In addition, eigenvalue centrality indices and several partitioning algorithms have their root in spectral analysis of graphs. Let a graph has an adjacency matrix A and let an n by n degree matrix D be defined as

$$d_{ij} = \begin{Bmatrix} d(i) & \text{for } i = j \\ 0 & \text{otherwise} \end{Bmatrix} \tag{11}$$

Then the Laplacian of the graph is defined as $L = D - A$. L is a symmetric positive semi-definite matrix. A normalized Laplacian is $\bar{L} = D^{-\frac{1}{2}} L D^{\frac{1}{2}}$.

The second smallest eigenvalue, λ_2, of the Laplacian is the algebraic connectivity

of a graph. If λ_2 is positive the graph is connected, and the eigenvector corresponding to it, u, adds up to zero [44]. The simplest way of bisecting a graph using the spectrum involves dividing the set of vertices into half using the median value of u and assigning vertices to the corresponding half. This bisection does not guarantee optimality [45]. There are various alternative cuts to bisection, such as ratio cut, sign cut and gap cut. Any two way partitioning can be recursively applied to find a k-way partitioning but [46] develops a spectral k-way ratio cut partitioning using k-way weighted quadratic placement and ratio cost metrics.

There are several important problems involving graphs such as finding the minimum spanning tree and determining whether two graphs are isomorphic or not. A *tree* is a connected acyclic graph. A spanning subgraph of G is a subgraph with vertex set $V(G)$. A spanning tree is a spanning subgraph that is a tree. The *minimum spanning tree* of a network is a spanning tree with minimum total length/cost. The most popular algorithms to solve this problem are Kruskal's algorithm [39] and Prim's algorithm [40].

Two graphs are isomorphic with each other if they contain same number of vertices connected in same way, i.e., if permutation of rows and columns of the adjacency matrix of one gives the other's adjacency matrix. This problem has a wider application in pattern recognition, biocomputing and finding network topologies in large networks. The classical solution for isomorphism problem is Ullmann's algorithm which does permutation of the main graph's or it's subgraphs adjacency matrix and compare it to the adjacency matrix of the second graph until a match is found or end of permutation [41]. Comparison of various subgraph isomorphism algorithms is found in [42]. Determining whether two graphs are

isomorphic or not is an NP complete problem.

2.3 Network Models

2.3.1 Random Networks

Random networks are types of networks whose underlying graph connections are governed by a random process. There are several models of random graphs, but the most prominent ones are Erdos-Renyi [47] and Gilbert [48] models proposed in the late 1960s. For a sufficiently large number of nodes, these networks exhibit a Poisson degree distribution. Erdos-Renyi random networks, $g_{n,m}$ are selected randomly from the set $G_{n,m}$ of all possible graphs that can be formed by n nodes and m links. A Gilbert random graph $g_{n,p}$ is composed of n nodes with pairwise connection probability p.

2.3.2 Small-world Networks

The small-world phenomenon has been studied in social networks since the early experimental works [49, 50] that led to the concept of "six degree of separation". Recent studies describe small-world networks as a network exhibiting high ordered locality (clustered) while still being globally small (short inter-node paths). Networks showing small-world property can be generated by interpolating a regular lattice and random network [51, 52]. The small-world property has been observed in author-collaboration networks, the world wide web (www) [53], the power-grid [51], and transport network [54].

Lattice networks are made from a point lattice, a regularly spaced array of points with finite dimension, and regular connections between points within defined range of lattice distance. Lattice networks are often described by the dimension of the point array (d), the set of points (V), and radius (r) [55, 56]. A node in a lattice network is connected

directly to all the nodes that are with in r distance from it. Lattice networks used for small-world network generation are usually either one dimensional (1-D) or two dimensional (2-D) with ring [51, 54] and torus [57] structures respectively.

The characteristic path length and the average graph clustering measure the global "smallness" and order in the network respectively. The difference in the clustering and characteristic path length values between the regular lattice and the random network determines whether a small-world network emerges due to rewiring or not. For instance, if several networks are constructed from a 1-D regular lattice by rewiring links randomly with probability $\mu = [0, 1]$, as shown in Figure 2.2, then the characteristic path length varies from an order of $|V|/r$ to $\ln|V|/\ln r$ and the clustering varies from approximately 0.75 to an order of $r/|V|$ [56]. Generally r is much smaller than $|V|$ and it has been shown that there is a broad range of μ that generates networks with large clustering, C, and small characteristic path length, $\bar{d}$. This is because $\bar{d}$ drops rapidly for low values of μ, and C does not start dropping until μ is significantly high [51].

There are some 2-D lattice networks on square grids that have been used to study small-world behavior [58, 59], but they are often not built on planar lattices. For road networks, which are often approximated as planar or near-planar with small average degree, these models are impractical because 2-D lattice networks have high average degrees for $r > 1$.

2.3.3 Scale-free Networks

Scale-free networks exhibit a power-law degree distribution [24, 52]. The probability that a node has k links, for a tunable scaling exponent γ, is $P(k) \sim k^{-\gamma}$ [60].

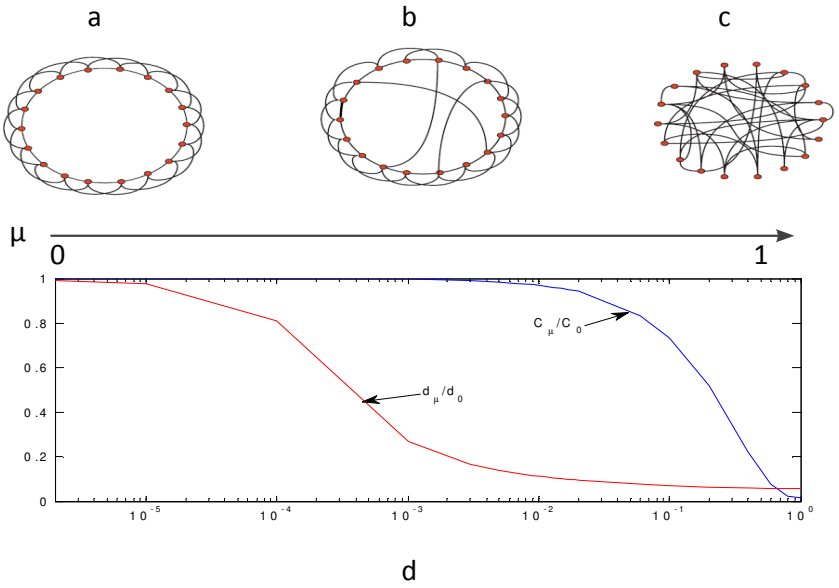

Figure 2.2. Watts-Strogatz model, rewiring of a regular lattice. (a) Regular 1-D lattice ($\mu = 0$) (b) Rewired network with $\mu = (0, 1)$ (c) Random network ($\mu = 1$) (d) Variation of graph clustering and characteristic path length.

The long-tailed distribution is often achieved using a growth and preferential attachment mechanisms that favor the connection of new nodes to nodes that have high degrees [56]. Scale-free networks are characterized by the existence of mega-hubs, as shown in Figure 2.3, and their tolerance to random failures. They also have, generally, a small diameter and their clustering coefficient decrease with an increase in network size.

2.4 Network Coarsening

Large and complex networks are common in many fields of studies; for instance, protein structures, the world wide web (www), road networks, etc. The large size of a network often presents a computational challenge on network based applications. The most common methodologies used to mitigate this computational burden are graph partitioning and coarse graining.

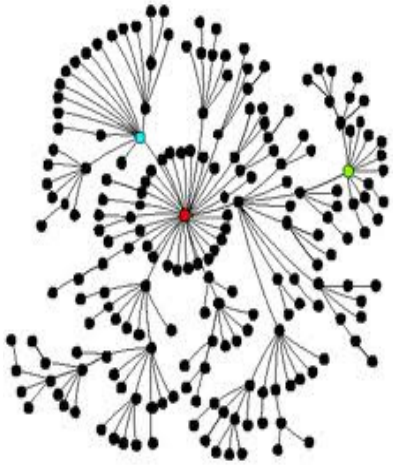

Figure 2.3. A scale-free network.

Graph partitioning algorithms divide the node set of a graph into disjoint subsets. A k-way partitioning divides the node set into k disjoint sets. The partitioning procedure usually will have a constraint in the form of the sizes of the partitions and a cost minimization objective function. The partitioning is accomplished by removing links. The sum of the weights of the removed links is often part of the partitioning cost. This problem does not have exact solution and hence is intractable in polynomial time. Partitioning a graph can help develop efficient solutions to a wide range of problems which include distributing computation [61] and community detection in social and biological networks [62] among others.

Coarse-graining is the process of reducing the resolution of a network. Coarse-graining is often achieved by iterative application of link/node contraction. A link is said to be contracted if the nodes at its opposite ends are lumped together [61], while nodes are contracted by replacing shortest paths passing through them with a shortcut [63]. In addition to complexity reduction, coarse-graining is important to implement region and

hierarchy-based applications, such as routing. A coarsening-based routing finds the union of local paths (paths inside partitions) and global paths (paths between partitions) [64, 65].

There are various heuristics used to solve the graph partitioning problem. They are classified as hypergraph-based versus graph-based, spectral versus iterative and k-way versus two-way [46]. The classical algorithm for partitioning [66] starts from an arbitrary two partitions and exchanges elements of the subsets that reduce the cost until a local minimum is reached. The two-way partitioning is extended to k-way partitioning by trying to achieve pairwise optimality.

Spectral partitioning equate the spectrum of a graph to a cost function. The eigenvector of the k smallest eigenvalue of the Laplacian of a graph can be used to find an embedding of the node into k-dimensional subspace which is transformed into k partitions [46]. Some partitioning algorithms use centrality measures, for instance, link-betweenness. This algorithm singles out the highly central links based on betweenness indices and remove them one by one until the graph is no longer connected [67].

The links and nodes to be contracted for coarse-graining are often selected based on some measure of importance. Another coarsening strategy is lumping all the node in close spatial proximity [6, 68, 69]. Part of the spectrum of the graph, for instance, the entries in the eigenvector corresponding to the largest eigenvalue, can also be used to select the nodes that are to be lumped [5]. Coarse-graining is mainly used to reduce computational complexity [70–72].

Graph-partitioning and coarse-graining are closely related and sometimes interdependent. For instance, coarse-graining is used in multilevel partitioning [61, 73]. Multi-

level partitioning generally involves three steps; constructing a coarse graph, partitioning the coarse graph, and uncoarsening and refining the partitions. Partitioning can also be used to implement coarse-gaining [74]. Another related concept to coarse-graining is clustering. Clustering in graphs is used to find subgraphs that are strongly connected [62, 75]. It is similar to partitioning and often one can be found from the other.

2.5 Survivability Measures

Survivability is the ability of a system to continuously deliver essential services and maintain essential properties such as integrity, confidentiality and performance despite attacks, failures or accidents that damage or compromise a significant portion of the system [1]. Attacks can be physical, such as the destruction of nodes and/or links, but can also include intrusions, probes and denial of service. Failures, which are mostly internally generated, include system deficiencies, software design errors, hardware degradation, human error and corrupted data. Accidents are usually random and external, such as natural disasters [1, 2].

In an early work [3] on the survivability of command, control and communication (C^3) networks, survivability was described in terms of the existence of communication paths, the number of nodes in the largest connected section, the shortest surviving paths, the fraction of nodes that can still communicate, and the maximum time required to transmit messages after attack. In Markovian models, network survivability is quantified by combining various performance models of the different failure propagation and recovery phases [4].

There are primarily two types of survivability measures, topology-based and traffic-

based. A weighted-average of the two can also be used as a survivability measure [76]. Topological measures are built on some notion of connectivity, such as the smallest number of disjoint paths between a node pair in the network. High connectivity in a network is generally an indication of survivability [9, 76–79]. The ratio between the number of node pairs that can still interact after a fault, and the number of node pairs that were able to interact before fault, is another topology-based survivability measure [76, 80].

Traffic-based survivability measures use the amount of disruption on the flow between origin-destination pairs to quantify survivability [81–83]. The disruption can be in terms of the demand not met [81], the traffic blocking probability at different sections of the network [83] and probability distribution of the percentage of total data flow after failure [82].

There are also methods that employ vulnerability analysis of the network components to quantify survivability [84, 85]. In these approaches, various attack scenarios are analyzed and often the survivability of the system is determined by its weakest component. This is more common in computer and information networks where the main attack types are intrusion and compromise. This approach does not take the response of the system into consideration once an attack has occurred. Hence, one cannot tell whether the system can provide essential services after attack or not.

Kang et al. [78] proposed measuring the survivability of a network by removing one node at a time until the network is disconnected. They defined the survivability as the sum of the connectivity of the networks obtained after each removal. A similar approach that removes links instead of nodes until the network is disconnected is found in [79].

Molisz [82] used probabilistic attack scenarios and assessed their impact on the delivered total data flow to measure the expected survivability. A survivability model and measure that addresses both structural and functional issues of a network is proposed by Heegard and Trivedi [4]. They measure the expected values of total loss and end-to-end delay in a telecommunication network using a continuous time Markov chain.

Graph-theoretic measures such as efficiency, clustering, assortativity and the likes are used to characterize and study structural performance and property of a network. Let d_{ij} be the number of edges (hops) along the shortest paths between nodes i and j of a graph, n the number of nodes in the graph and V be the node set, then efficiency of the graph is defined as [54]

$$\eta = \frac{1}{n(n-1)} \sum_{i \neq j \in V} \frac{1}{d_{ij}} \tag{12}$$

Assume that a measure, M, characterizes a desired network performance, and that it is equal to M_0 and M_a before and after a fault, respectively. Then, the survivability of the network can be quantified as some statistic of the difference, ΔM, between M_0 and M_a. Alternatively, the ratio of the difference to the original performance, $\Delta M/M_0$, can be used. In addition, the time needed to restore the performance to an acceptable value, t_r, can be used to quantify the survivability [4].

Examples of performance measures of a network that are used in survivability analysis include: (a) The shortest path length (communication delay, transit time) between origins and destinations, (b) The network flow (transfer rate, packet loss and blocking probability) from origins to destinations, and (c) The connectivity of given node-pairs. The

connectivity of network, $\gamma(k)$, after k nodes are removed can be defined as [78]

$$\gamma(k) = \sum_{i=1}^{n-k-1} \sum_{j=i+1}^{n-k} \sum_{p=1}^{m_{ijk}} \frac{1}{l(p)}, \tag{13}$$

where, m_{ijk} is the number of disjoint paths between nodes i and j after k nodes are removed, n is the number of nodes in the original network and $l(p)$ is the length of the p^{th} disjoint path between i and j. Then, the connectivity-based survivability, S, is

$$S = \sum_{k=1}^{v-1} \gamma(k), \tag{14}$$

where v is the maximum number of nodes that can be removed before the network is disconnected. This measure captures the structural behavior of a network as individual nodes are randomly removed but fails to indicate the functional survivability of the network.

CHAPTER 3

NETWORK FAULTS AND ROBUSTNESS

Robustness of a network is mainly reflected by its ability to maintain its characteristic behavior under perturbation. Network perturbations include single or multiple random failure of nodes and/or links, targeted attacks and large-scale failures. In general, robust networks are survivable. The behavior of a network is often described by some form of the measures described in section 2.2. Thus, robustness can be measured by, for instance, the increase in characteristic path length and/or the increase in the number of connected components.

This chapter presents a linear programming based approach for finding the average connectivity of a network. Then the average connectivity and network efficiency are used to analyze the response of a network to different types of perturbation. The two measures are selected because they complement each other. While efficiency measures the compactness of the network, small inter-nodal distances, connectivity measures availability of optimal alternative paths. In addition to the disruption in the networks behavior, the size of fault the network can withstand before a significant breakdown in behavior happens is also used to measure the network's robustness.

3.1 Average Connectivity

Connectivity is one of the most important measures of network survivability. Existence of a path between network elements is crucial to the normal functioning of any network. The classical definition of connectivity of a network is the minimum number of nodes or edges whose removal will increase the number of connected components in the

network. This definition is equivalent to the minimum number of disjoint paths between over all node-pairs in the network. This does not reflect the overall network property, since it is based on the worst case. For instance, the two graphs in Figure 3.1 both have a network connectivity of 1, although the one on the left is almost a complete graph.

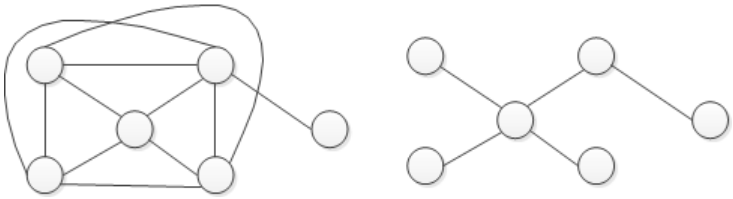

Figure 3.1. Graphs with equal network connectivity.

The algebraic connectivity of a graph, the second-smallest eigenvalue of the Laplacian matrix of the graph ($\lambda_2(G)$), is an alternative connectivity measure. But $\lambda_2(G)$ is upper bounded by the vertex connectivity of a graph, which in turn is upper bounded by the edge-connectivity. Therefore, the algebraic connectivity measure has similar drawback as vertex and edge connectivity measures.

To address this misrepresentation, an average connectivity measure was proposed [86]. As its name suggests, the average connectivity of a network is defined as the average of the vertex/edge connectivities over all node-pairs in the network. Let $k(u, v)$ be the minimum number of node removal that makes v unreachable from u, then the classical network connectivity, γ^c, is

$$\gamma^c = \min_{(u,v)\in V} k(u, v) \tag{15}$$

and average connectivity of the network, γ^a, is

$$\gamma^\alpha = \frac{1}{n(n-1)} \sum_{(u,v)\in V} \gamma(u, v). \tag{16}$$

The average connectivity measure in Equation (16) measures only the existence of disjoint paths without any regard to the quality of the paths. Another average connectivity measure built from the length of the disjoint paths is proposed in [78]. It is obtained by setting k to 0 in Equation (13). Thus, the average connectivity of a network, γ^l, is

$$\gamma^l = \frac{1}{n(n-1)} \sum_{(u,v)\in V} \sum_{p\in P_{uv}} \frac{1}{l(p)} \tag{17}$$

where P_{uv} is the set of the shortest disjoint paths between nodes u and v and $l(p)$ is the length of path p.

A complete graph has $\gamma^c = \gamma^a = n-1$, $\lambda_2 = n$ and $\gamma^l = \frac{n}{2} - 1$. From the three connectivity measures discussed here, only γ^l evaluates both the number and quality of disjoint paths. Table 3.1 shows that γ^l is the most intuitive and informative measure of the networks characteristics in terms of connectivity. Therefore, this work will use γ^l as a measure of average connectivity of a network. Hence, average connectivity, γ, will refer to γ^l in the rest of this chapter.

Table 3.1. Summary of connectivity measures of graphs in Figure 3.2

Measures	a	b	c	d
γ^c	1	1	0	1
λ_2	1	1	0	0.27
γ^a	3	1	2	1
γ^l	1.87	0.67	1.22	0.58

To find the average connectivity of a network, the set of shortest disjoint paths between every pair of nodes needs to be found. The simplest algorithm to find P_{uv}, for nodes u and v, is a greedy algorithm that can be implemented by iterating any shortest path algorithm. The algorithm first finds the shortest path, then removes all edges in the shortest path and run the shortest path algorithm again. This is repeated until the two nodes are no

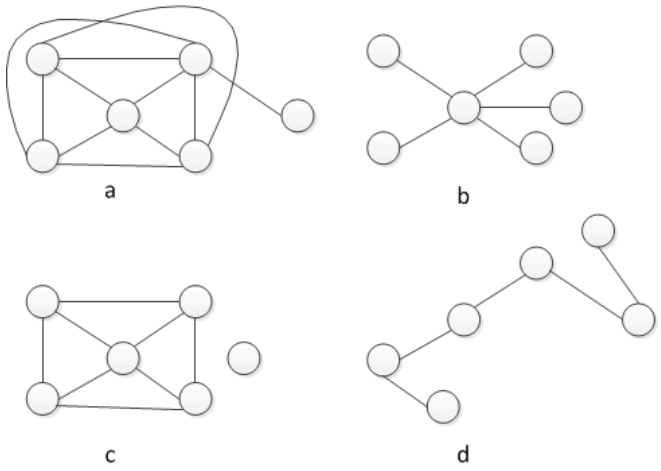

Figure 3.2. Graphs with varying structures.

longer connected. A greedy algorithm that tries to minimize the total cost of edges used to form edge-disjoint paths set can be summarized with the steps shown in Figure 3.3.

```
Input: graph
Output: set of paths
P = greedy (G)
initialize disjoint path set to empty (P← {})
STEP 1: find the shortest path (p) between i and j in G
    if p is empty
        return
    else
        go to step 2
STEP 2: remove all edges that belong to p from G
    Add p to P
    go to step 1
```

Figure 3.3. Greedy shortest disjoint paths algorithm.

The greedy approach does not always yield an optimal solution. For example, for the graph shown in Figure 3.4, the greedy algorithm returns the path ABDF between nodes A and F in the first run and finds no other path in the next iteration since the removal of edge AB and DF disconnects A and F. But, it can be shown that there are two edge-disjoint

paths, ABCEF and AJIDF, between A and F.

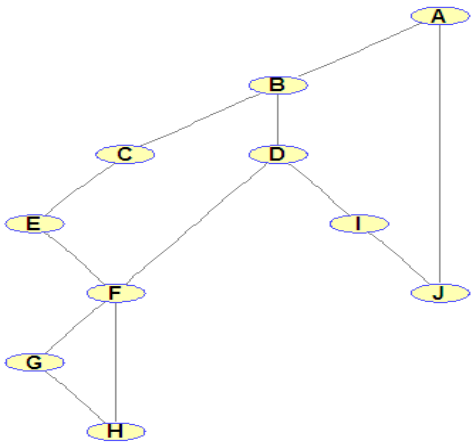

Figure 3.4. A simple graph showing the shortfall of greedy algorithm.

One alternative to the greedy algorithm is a linear programming approach that is guaranteed to return the optimal set of shortest disjoint paths. Finding the optimal set of disjoint paths between two nodes is achieved in two steps. First, the number of the maximum achievable disjoint paths is found. Second this number is incorporated as a constraint to an integer linear program (ILP) that minimizes the total number of edges needed to achieve the maximum number of disjoint paths.

3.1.1 Flow and edge-disjoint Paths

For a pair of vertices i and j, the terminal capacity from i to j, is the value of the minimum $i - j$ cut. If all arcs of a graph have a unity flow-capacity then the terminal capacity of i and j is the minimum number of arcs that must be removed to disconnect i and j, which is equivalent to the number of edge-disjoint paths between i and j [17]. Since a minimum cost requirement is not associated with the min-cut/max-flow optimization, the length of the paths returned is not guaranteed to be optimal. Thus, given an optimal terminal

capacity, k^*, an LP can be formulated to minimize the number or cost, in case of weighted graphs, of links used to meet the k^* edge-disjoint path requirement. This is a min-sum type of minimization described in section 2.2.3.

Let I be the node-link incidence matrix, where $I(i,j)$ is 1, -1 or 0 if node i is the origin, destination or neither of link j respectively. Let x_j be the binary decision variable whether link j is used or not in obtaining k^*, and w_j be the length/cost of link j, then the min-sum edge-disjoint paths can be found by solving

$$\min \sum_{j \in E} w_j x_j \text{ subject to,} \tag{18}$$

$$\sum I(i,j)x_j \quad = \quad \begin{cases} k^*, & i = s \\ 0 & \text{otherwise} \\ -k^* & i = t \end{cases} \tag{19}$$

Equation (18) is the objective of the ILP. The constraints in Equation (19) are the number of disjoint path requirement from s to t. If the graph is not weighted, w_j is equal to one for all edges. In this ILP there are $|V|$ constraints and $|E|$ decision variables. The edge-disjoint paths are obtained by grouping the set $\{j \in E : x_j = 1\}$.

The max-flow and min-sum LP used to find the min-sum edge disjoint paths can be further simplified into a single ILP. Let $G'(V', E')$ be a graph formed by adding two additional nodes each attached with a single edge, that has infinite capacity and negative cost, to the source and sink nodes of G as shown in Figure 3.5. Then, assuming the magnitudes of the weights of (s', s) and (t, t') are very large but finite, the min-sum edge-disjoint path set can be constructed from the minimum cost flow pattern from s' to t'.

Let the weight of edges (s', s) and (t, t') be $-M$, where $M >> 1$, x_j be a decision variable associated with edge $j \in E'$. If the objective of the optimization is minimizing

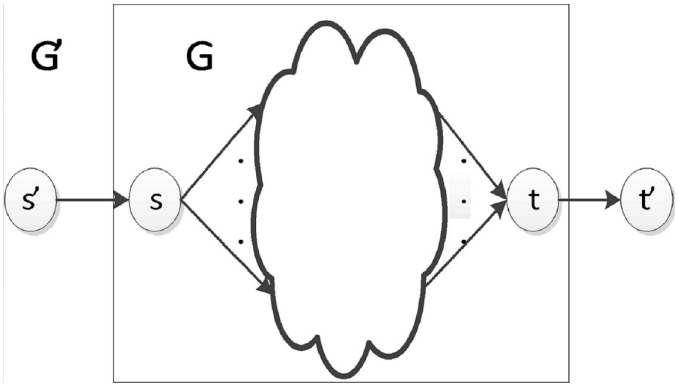

Figure 3.5. Graph transformation to find the optimal set of disjoint paths.

$\sum_{j \in E'} w_j x_j$, then an increase in $x_{(s',s)}$ decreases the objective function. On the other hand, it is a decrease in all $x_j, j \in E : j \neq (s',s),(t,t')$ that contributes to the minimization. Hence, the large value of M secures that the number of disjoint paths is given priority than the length of the paths. If M is not made large enough the program will converge to the trivial solution of $x_j = 0, \forall j \in E$.

Since the number of disjoint paths between two vertices is always less than or equal to the minimum of the degrees of the vertices, $x_{(s',s)}$ is bounded from up by the minimum of the outgoing degree of s and the incoming degree of t. From flow conservation, $x_{(s',s)} = \sum x_{(s,i)}$ where $i \in V$ and $A(s,i) > 0$. Since x_j is an integer for all $j \in E$, then $x_{(s',s)}$ is also an integer. In fact, $x_{(s,s')}$ is the number of edge-disjoint paths. Let I' be the vertex-edge incidence matrix of graph G', then a single mixed integer linear program that can find the set of edges that can be used to construct the min-sum edge-disjoint path set is

$$\min \sum_{j \in E} w_j x_j - M x_{(s',s)}, \text{ subject to} \tag{20}$$

$$\sum_{j \in E'} I'(i,j) x_j = 0, \tag{21}$$

$$0 \leq x_{(s',s)} \leq \min(\delta^+(s), \delta^-(t)). \tag{22}$$

where $\delta^+(i)$ and $\delta^-(i)$ are the outgoing and incoming degrees of vertex i respectively.

3.1.2 Vertex-disjoint Paths

Similarly to edge-disjoint paths, the problem of finding vertex-disjoint paths can be formulated using standard LPs. Let P be list of edges in a path, u, v and z be vertices in the graph, $u, v, z \neq i, j$. For two paths, P_1 and P_2, between vertices i and j, to be vertex-disjoint, if (u, v) is in P_1 then (z, v) can be in neither P_1 nor P_2. This implies that vertex-disjoint paths are also edge-disjoint. Therefore, the maximum number of vertex-disjoint paths can be found by appending this constraint to the max-flow constraints. In other words,

$$\max k, \text{subject to} \tag{23}$$

$$\sum_{j \in V} f(i,j) - \sum_{j \in V} f(j,i) = \begin{cases} k\,, & i = s \\ 0 & \text{otherwise} \\ -k & i = t \end{cases}, \tag{24}$$

$$\sum_{j \in V} f(i,j) \leq 1,\ i \neq s, \tag{25}$$

$$1 \geq f(i,j) \geq 0\ \ (i,j) \in E. \tag{26}$$

Equation (25) indicates that no vertex can be used more than once in the paths between i and j. Let I^v be the outgoing vertex-edge incidence matrix, where $I^v(i,j)$ is 1 if node i is origin of link j and 0 otherwise. To minimize the total cost of edges used, an ILP is formulated as

$$\min \sum_{j \in E} w_j x_j \tag{27}$$

$$\sum_{j \in E} I(i,j) x_j \leq \begin{cases} k* & i = s \\ 0 & \text{otherwise} \\ -k* & i = t \end{cases}, \tag{28}$$

$$\sum_{j \in E} I^v(i,j) x_j \leq 1,\ i \neq s \tag{29}$$

Both the maximization and minimization programs have $2|V|$ constraints and $|E|$ decision variables. Hence, this approach will get computationally intensive as the size of the graph increases. If all arc capacities and supplies/demands of vertices are integers, then an integer solution exists for the minimum cost flow problem [87]. Thus, all the ILPs formulated to find edge-disjoint and vertex-disjoint paths are guaranteed to return an optimal solutions.

3.1.3 Algorithm Performance

The greedy algorithm shown in Figure 3.3 is used as a benchmark to evaluate the performance of the LP approach to finding disjoint paths. Performance measurement is only done for edge-disjoint paths, since the conclusions about the algorithms are easily extensible to the vertex-disjoint path cases. The number of disjoint paths obtained, the total length/cost of the paths (for cases where both algorithms return equal numbers of disjoint paths), and the running time of the algorithms are all used for performance comparison.

For numerical simulations, undirected random Gilbert graphs [48] are generated with size connection probability of $\frac{\log|V|}{|V|}$, where $|V|$ ranges from 10 to 100 incrementing by 10. To solve the ILPs, MATLAB's *bintprog* from the optimization toolbox is used. Edge-disjoint shortest paths are computed for all combinations of vertices. Hence, there are $|V||V-1|/2$ runs for each algorithm and for each graph. Figure 3.6a shows the time cost of the ILP approach relative to the greedy algorithm, and Figure 3.6b shows the number of vertex-pairs whose cost of edge-disjoint path set is improved by ILP.

From the simulation results, it can be seen that as the network size increases LP improves a significant number of paths in length. In addition, the number of edge-disjoint

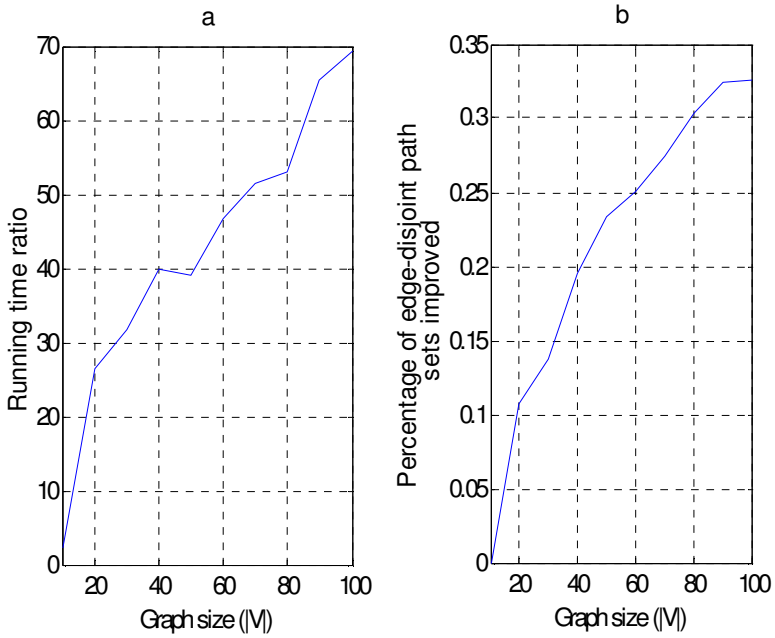

Figure 3.6. Peformance comparison of greedy and LP algorithms (a)LP : Greedy running time ratio (b) Percentage improvement on length of paths using LP

paths is improved for several node-pairs. But the percentage improvement does not show any pattern with the size of the network, which might be attributed to the random nature of the graph generation strategy.

3.2 Network Response to Perturbation

3.2.1 Single node failures

A single node/link failure occurs in one of two situations: either a random node/link in the network is faulty, or there is a targeted attack on the most important node of the network. In this section only node failures are considered for the sake of computational simplicity, but by extension it can be easily shown that link failures have similar property as node failures. The response of a network to failures is quantified by the amount of relative loss, $\Delta M/M$, due to the failure as discussed in section 2.5. The measures used are

connectivity (γ) and efficiency (η). Network efficiency is computed using Equation (12).

The average response of scale-free and Erdos-Renyi random networks for a single node failure in 10 runs is shown in Figures 3.7 and 3.8. The plots show centrality versus relative loss of network average connectivity and efficiency, 0 indicates the least central and 1 indicates the most central nodes of the network.

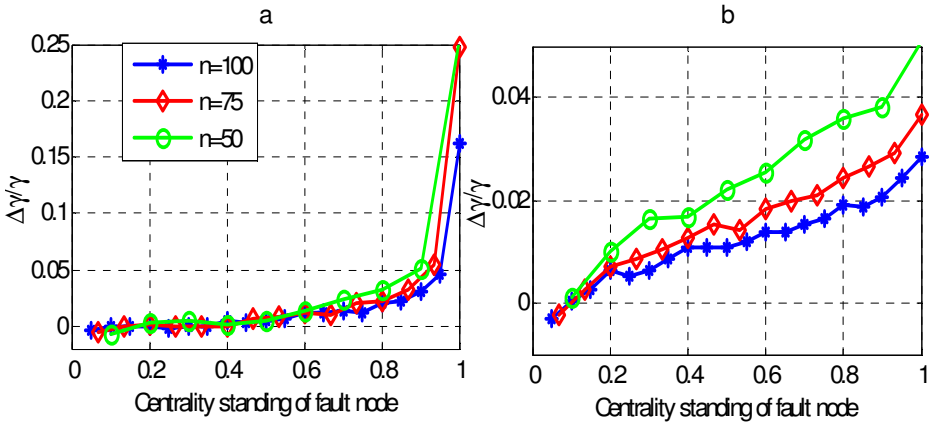

Figure 3.7. Relative loss in connectivity in (a) scale-free and (b) random network.

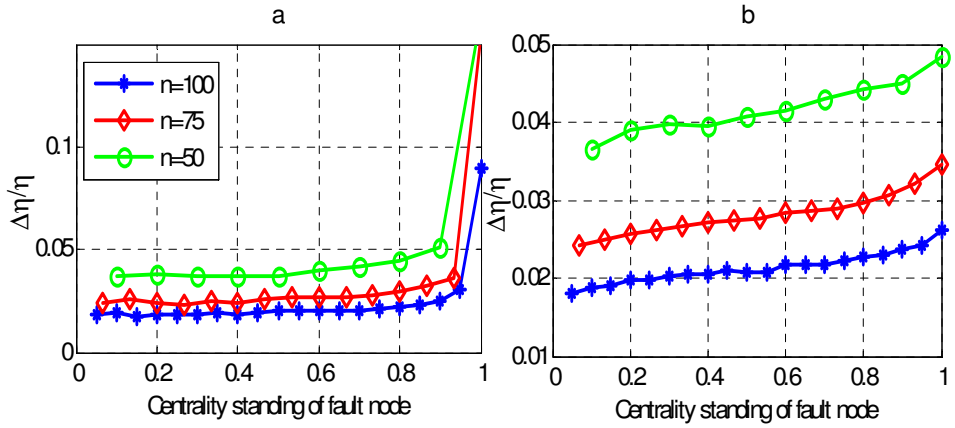

Figure 3.8. Relative loss in efficiency in (a) scale-free and (b) random network.

The results confirm that scale-free networks are vulnerable to targeted attack, i.e., failing of highly central nodes. In addition, comparison of Figures 3.7a and 3.8a suggests

that connectivity is more sensitive to node failures than efficiency in scale-free networks. Figures 3.7 and 3.8 also suggests that an increase in the size of the network dampens the effect of single node failures, which can be intuitively explained as "safety in numbers".

3.2.2 Multiple Failures

Multiple failures, similar to single node failures, can result from either random failures or from targeted attacks. In addition, the failures can be confined to a certain part of the network or can be distributed all over the network. Hence, this work considers four types of multi-failure scenarios: random distributed failures, random localized failures, targeted distributed failures (coordinated attacks), and targeted localized failures. In this section, only random and targeted distributed failures are presented and the discussion on the other two is deferred to the next section.

Figure 3.9 shows an average relative loss in average connectivity and in efficiency when 5 nodes from a network of 100 nodes are removed. The result shown is an average over 10 random scale-free and Erdos-Renyi networks.

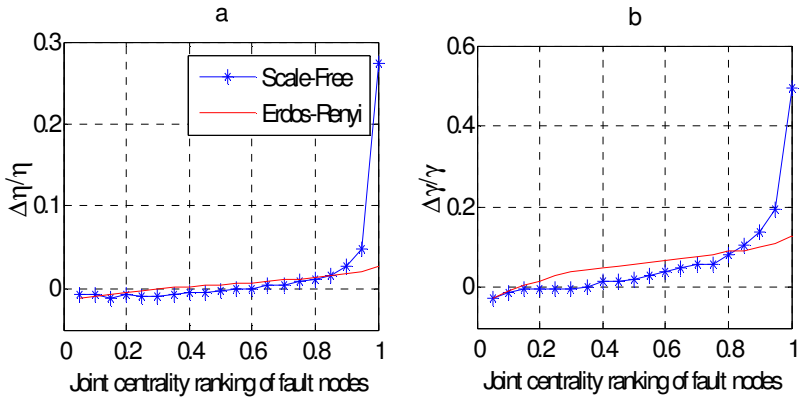

Figure 3.9. Average response to multiple node failures (a) loss in efficiency (b) loss in connectivity

Since the losses in scale-free networks are smaller than the losses in random networks for the most part of plots in Figure 3.9, it can be concluded that scale-free networks are more robust than random networks under random perturbations. Targeted failure (coordinated attacks) are shown to have a severe impact on scale-free networks. Targeted attacks in this experiments were simulated by removing the 5 nodes with the highest sum total centrality index. In general, the loss patterns in multi-node failure are similar with single node failures except for the size of the losses.

3.3 Localized Faults in Spatial Networks

Let n_f be the number of fault nodes, then there are $\binom{n}{n_f}$ fault scenarios. Hence, all possible fault scenarios cannot be considered for even moderate sized networks. Most cases of failures in an infrastructure network, such as the ones caused by weapons of mass destruction and natural disaster, damage nodes and links in a shared neighborhood. Since for every node in a network there is only one set of neighborhood with size n_f, the assumption in the locality of fault will reduce the number of fault scenarios to less than or equal to the number of nodes. Locality can refer to graph locality (topological neighborhood) or geographic locality (physical neighborhood). This work will focus on the latter, since infrastructure networks can generally be assumed to be geographically embedded.

3.3.1 Fault Scenarios

Let r be the radius of an area in a network, G, that is destroyed. Then assuming a random fault centered at a node, $v \in V$, the group of nodes that are destroyed depends on the location of v in the network. Hence for each node a neighborhood can be built as shown in Figure 3.10. In this figure, a weight on a link of the graph is the geometric

distance between its end nodes. Each neighborhood represents a possible localized fault scenario of radius r.

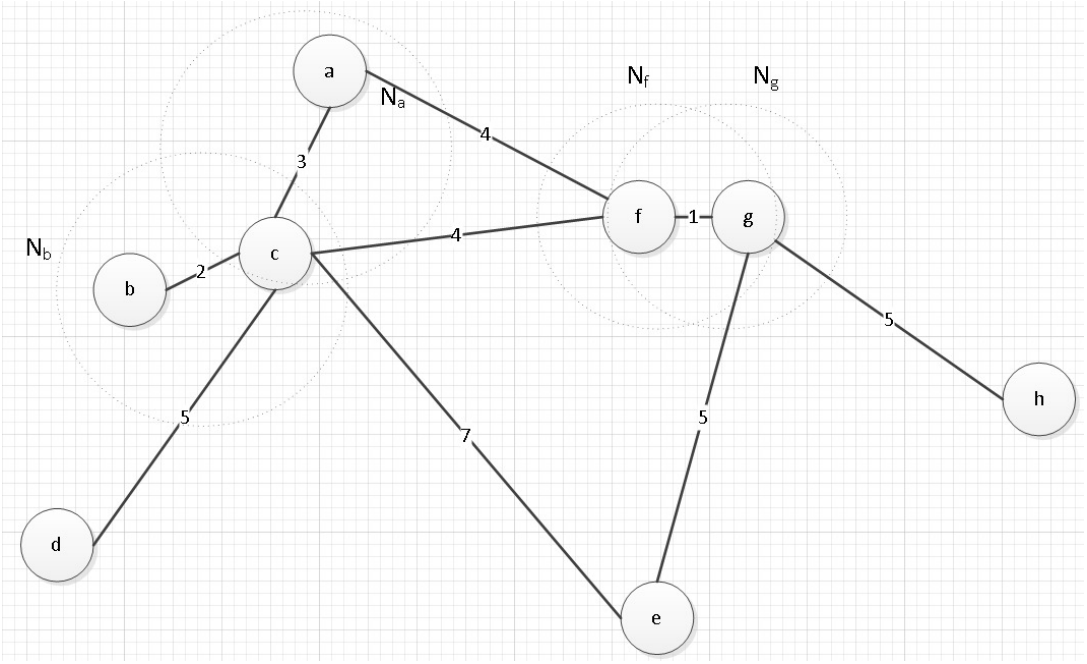

Figure 3.10. Construction of neighborhood network for $r = 3$.

Let N_v denote the neighborhood v, i.e., the set of nodes that are within r distance from node v. Then, the possible fault scenarios can be represented by a $|V|$x$|V|$ boolean matrix, N, such that $N(i, j)$ is 1 if node j belongs to neighborhood i. Figure 3.10 shows that neighborhoods N_f and N_g are identical. Hence, the actual fault scenarios are 7 instead of 8, which is the number of nodes in the network. The number of overlapping scenarios increases as the size of the network and radius of the fault increase. Therefore, there are much less localized multiple-fault scenarios than single-fault scenarios.

For both single node failures and distributed multiple failures, it is shown that the centrality of the fault is an important predictor of the loss suffered by the network due

to the fault. To extend a similar approach, it is important to define the centrality of a neighborhood. A simple way to define centrality of a neighborhood is aggregating the centrality of the nodes in it. Let $C(v)$ be the centrality index of v, then the centrality of N_v is $C^N(v) = N_v C^T$. Thus,

$$C^N = NC^T \tag{30}$$

The centrality of a neighborhood is intended to evaluate the importance of the neighborhood towards the rest of the network. But, the importance measure of a node inside a neighborhood is not limited to those outside the neighborhood. Hence, a simple aggregation will not only measure the importance of a region to other regions, but also to itself.

Let $D(i,j)$ be the set of shortest edge-disjoint paths between nodes i and j and let $D^v(i,j)$ be the subset of $D(i,j)$ passing through v. Similar to the betweenness centrality in Equation (9), the centrality centrality of a neighborhood can be computed as

$$C_{N_v} = \sum_{(i,j)\in V\setminus N_v} \frac{|D^{N_v}(i,u,j)|}{|D(i,j)|} \tag{31}$$

where $D^{N_v}(i,u,j) \in D(i,j)$ that contains at least one node $u \in N_v$.

3.3.2 Neighborhood Centrality vs. Loss

A numerical experiment is conducted on a portion of the North Carolina highway network around the Piedmont Triad region, shown in Figure 3.11. For the sake of simplicity, only node failures and connectivity-based robustness measure are considered, but link failures can also be simulated by placing an intermediate failed node on the failed link. This experiment is used to demonstrate that the centrality of a neighborhood impacts the network response to the failure of that neighborhood. In addition, the size of failure and the worst expected loss in average network connectivity are shown to be directly related.

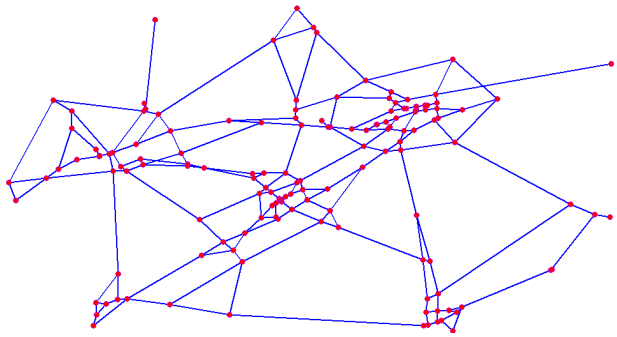

Figure 3.11. The simplified highway network of the Piedmont Triad.

For a given fault radius, a neighborhood network is first constructed as described above. Then the group centrality of each of the neighborhoods is computed. Figure 3.12a shows the expected loss is generally increasing with the centrality of the neighborhood removed. The worst loss clearly happens when the most central neighborhood is removed. Figure 3.12b shows the worst expected losses as a function of the size of fault in terms of the average radius of the neighborhoods. The worst loss is assumed to happen when the most central neighborhood is removed. As expected, the loss increases almost linearly with the size of fault.

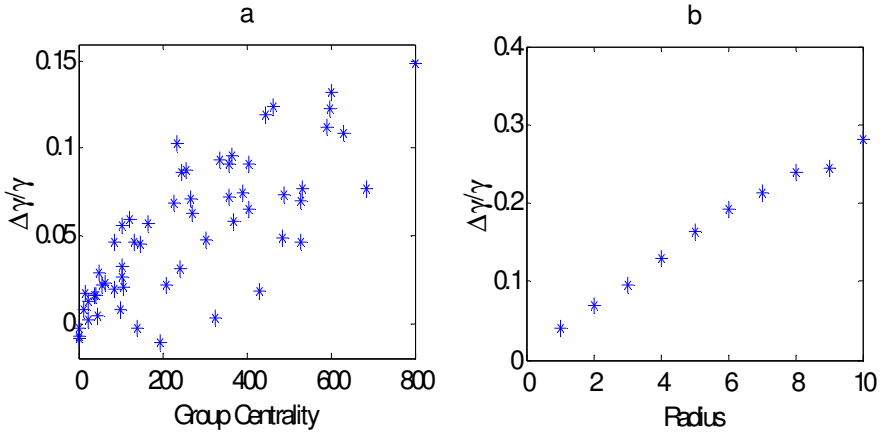

Figure 3.12. (a) Centrality versus connectivity loss. (b) Fault size versus loss.

CHAPTER 4

NETWORK LOCAL SURVIVABILITY MEASURES

In the preceding chapter and in much of the reported research on network survivability, the survivability of the network is described as a disruption on the average characteristic behavior of the network. These approaches underestimate the fact that a fault is not expected to impact all parts of the network equally. Due to the large size of most real-world networks, averaging does not reflect actual impact of the faults, since impacts of some faults are contained in local neighborhoods while others have a global reach.

This chapter analyzes the local aspects of survivability by quantifying the propagation of loss in the network in terms of the distance of the fault from the shortest path connecting a given origin-destination node pair. Two performance measures, connectivity and maximum feasible flow, are used to quantify both the local and global survivability of the network.

4.1 Distance between a node and a path

The failure of a node in a network impacts individual nodes (degree, clustering coefficient), as well as interaction between nodes (shortest path, max-flow, connectivity), and the entire network characteristics (characteristic path, connectivity, average clustering, ...). In large networks, measuring the impact of a node failure on the entire network can be misleading about the survivability of the network because the quantifiable impact will be dampened due to the network size. In addition, intuition suggests that nodes in close proximity are more dependent on each other than nodes that are further apart.

For instance, a fault on node h in Figure 4.1 increases the length of the shortest path

between nodes a and A by 3 hops and reduces the number of disjoint paths from 2 to 1. On the other hand, a fault on node h does not affect the shortest path from b to B but reduces their number of disjoint paths by 1. Both the shortest path and number of disjoint paths between c and C are not affected by the fault on node h.

It is no accident that the failure of node h in Figure 4.1 affects node pairs (a, A), (b, B), and (c, C) differently. The fault node is in the shortest path of (a, A), 1 hop away from shortest path of (b, B), and 3 hops away from the shortest path of (c, C). Thus, the impact of the failure of node h on the node pairs is positively correlated to the distance between h and the shortest path of the node pairs. Therefore, a survivability measure that is a function of the distance of a node from a node-pair will be more descriptive of the impact of node failure on the network.

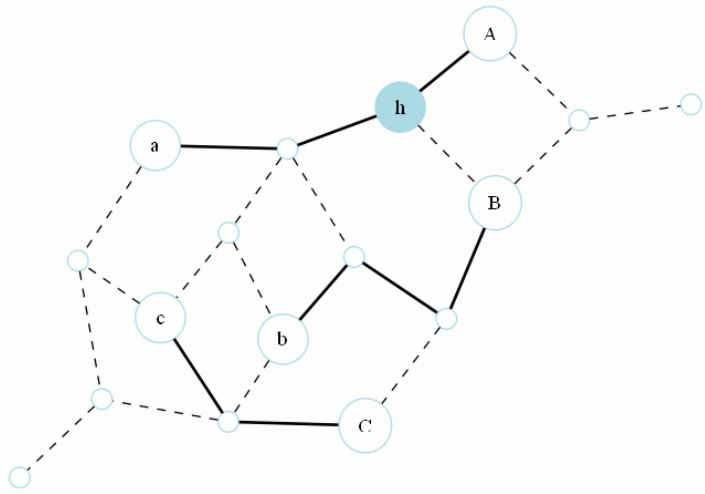

Figure 4.1. Diagramatic representation of the distance from a node pair to a node.

An intuitive definition for the shortest distance of a node to a path is the shortest distance between the node and any of the nodes in the path. The continuous form of this

distance is the length of a perpendicular line segment from a point to a line. Let P_{ij} be the list of intermediate nodes in the shortest path of nodes i and j, $i \neq j$, and l_{ij} be the shortest path length (SPL) between nodes i and j. Then for any node h, $h \neq i, j$, the shortest distance, d_{ij}^{h*}, from a node, h, to the shortest path, P_{ij}, is

$$d_{ij}^{h*} = \min_{k \in P_{ij}} l_{kh}. \tag{32}$$

Using Equation (32) in large networks is computationally expensive. In addition, Equation (32) is less representative of the fault node's significance to the node-pair. Therefore, a more robust distance measure which is less computationally expensive is proposed below. An approximation of the shortest distance between a node, h, and the shortest path between i and j (DNSP) can be suggested by the triangular inequality of triangle ihj, i.e.,

$$d_{ij}^{h} = l_{ih} + l_{hj} - l_{ij}. \tag{33}$$

Note that if h belongs to the shortest path between i and j, then $d_{ij}^{h} = d_{ij}^{h*} = 0$. Since the sum of two sides of a triangle is always greater than the third, $l_{ij} \leq l_{ih} + l_{hj}$. Hence, as suggested by the triangle inequality, $d_{ij}^{h} \geq 0$ for all i, j, h. This is proven below.

Theorem 2 In an undirected, graph d_{ij}^{h} is tightly bounded by 0 from below and $2d_{ij}^{h*}$ from above.

Proof. Let k be the node in P_{ij} that is closest to h. Hence from Equation (32), $d_{ij}^{h*} = l_{kh}$. Substituting $l_{ij} = l_{ik} + l_{kj}$ in Equation (33) leads to

$$d_{ij}^{h} = l_{ih} - l_{ik} + l_{hj} - l_{kj.} \tag{34}$$

Moreover, since l_{ih} is the length of the shortest path from i to h a path from i to h through k cannot be shorter, i.e.,

$$l_{ih} \leq l_{ik} + l_{kh}. \tag{35}$$

Similarly,

$$l_{hj} \leq l_{hk} + l_{kj}. \tag{36}$$

Substituting $l_{ih} - l_{ik} \leq l_{kh}$ from Equation (35) and $l_{hj} - l_{kj} \leq l_{hk}$ from Equation (36) into Equation (34) gives $d_{ij}^h \leq 2l_{kh} = 2d_{ij}^{h*}$.

The lower bound can be established from the fact that $l_{ih} + l_{hj}$ can never be less than l_{ij}, because l_{ij} is the length of the shortest path between i and j. When h is on the shortest path $l_{ih} + l_{hj} = l_{ij}$, and therefore $d_{ij}^h = 0$. Figure 4.2 shows the case where $d_{ij}^h = 2d_{ij}^{h*}$. $l_{ij} = 3$, $l_{ih} = 4$, $l_{jh} = 3$. Therefore $d_{ij}^h = l_{ih} + l_{hj} - l_{ij} = 4$, but $d_{ij}^{h*} = l_{kh} = 2$. Thus $d_{ij}^h = 2d_{ij}^{h*}$. ■

The above theorem suggests that d_{ij}^h is a reasonable approximation for a distance from a node to the shortest path of a node-pair.

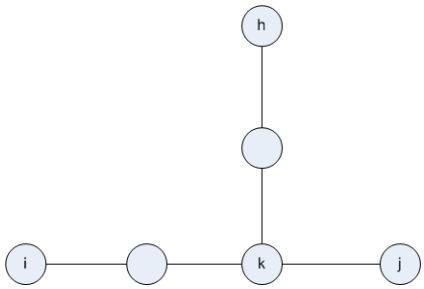

Figure 4.2. A case of $d_{ij}^h = 2d_{ij}^{h*}$.

The characteristic path length of a network is

$$\overline{l} = \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} l_{ij}. \tag{37}$$

Similarly, the average DNSP, $\overline{d}$, over all i, j and h, is

$$\overline{d} = \frac{1}{n(n-1)(n-2)} \sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} \sum_{h=1, h \neq i,j}^{n} d_{ij}^h. \tag{38}$$

Theorem 3 In an undirected graph, the average DNSP is equal to the average SPL (char-

acteristic path length) of the network.

Proof. Substituting Equation (33) into Equation (38)

$$\overline{d} = \frac{\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n}\sum_{h=1,h\neq i,j}^{n}(l_{ih}+l_{hj}-l_{ij})}{n(n-1)(n-2)}$$

Expanding the first term inside the bracket gives $\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n}\sum_{h=1,h\neq i,j}^{n} l_{ih}$

$$= \sum_{i=1}^{n}\left\{\sum_{j=1}^{n}\left\{\sum_{h=1,h\neq i}^{n} l_{ih} - l_{ih_{|h=j}}\right\} - \sum_{j=i}^{i}\left\{\sum_{h=1,h\neq i}^{n} l_{ih} - l_{ih_{|h=j}}\right\}\right\}$$

$$= \sum_{j=1}^{n}\left\{\sum_{i=1}^{n}\left\{\sum_{h=1,h\neq i}^{n} l_{ih} - l_{ij}\right\} - \sum_{i=1}^{n}\left\{\sum_{h=1,h\neq i}^{n} l_{ih} - l_{ij_{|j=i}}\right\}\right\}$$

$$= nn(n-1)\overline{l} - \sum_{j=1}^{n}\sum_{i=1}^{n} l_{ij} - \sum_{i=1}^{n}\sum_{h=1,h\neq i}^{n} l_{ih}$$

$$= nn(n-1)\overline{l} - 2n(n-1)\overline{l}$$

$$= n(n-1)(n-2)\overline{l}.$$

The second and third terms can be shown to be equal to the first by exchanging summation variables. Thus, the sum of the terms in the bracket is $n(n-1)(n-2)\overline{l}$ and hence, $\overline{d} = \overline{l}$. ■

4.2 Loss propagation

In this work, topology-based (edge-connectivity) and traffic-based performance measures (max-flow) are used to measure survivability. Losses reflected in these performance measures are used to quantify survivability. Let M_{ij} be a performance measure associated with nodes i and j and let M_{ij}^{h} be the corresponding measure after the failure of node h, then the relative performance loss between i and j due to the failure of h is denoted

$$\partial M_{ij}^{h} = \frac{M_{ij} - M_{ij}^{h}}{M_{ij}}. \tag{39}$$

The quantity M_{ij} can be solved for through a maximization problem, individually for each pair (i, j). Intuition suggests that both the edge-connectivity and max-flow cannot be improved by failure of any link or node in the network. This implies that M_{ij}^h is always less than or equal to M_{ij}. Therefore, $\partial M_{ij}^h \in [0, 1]$, with 0 being no loss and 1 being complete loss.

The severity of ∂M_{ij}^h is assumed to be related to the DNSP of h and (i, j). Therefore, the average relative performance loss (ARL), between any node pair that is at a DNSP of d away from fault node, is

$$L_D(d, h) = \frac{1}{|I_d^h|} \sum_{i,j \in I_d^h} \partial M_{ij}^h, \tag{40}$$

where $I_d^h = \{(i, j) \in V^2 \text{ such that } d_{ij}^h = d\}$.

To quantify the ARL between node pairs as a function of their DNSP from the fault, an additional measure, $L_D(d)$, needs to be introduced. $L_D(d)$ is obtained by averaging $L(d, h)$ over all fault cases. Let $p(h)$ be the probability of failure of node h, then the ARL due to a single node failure is

$$L_D(d) = \sum_{h=1}^{n} p(h) L(d, h). \tag{41}$$

In this chapter only single-node failures are considered, i.e., $\sum p(h) = 1$. If the probability of failure is assumed to be equal for all nodes, i.e., $p(h) = 1/n$, $\forall h \in V$, then, Equation (41) can be rewritten as

$$L_D(d) = \frac{1}{n} \sum_{h=1}^{n} L(d, h). \tag{42}$$

Since $L(d, h)$ and $L_D(d)$ are obtained by averaging several ∂M_{ij}^h, the range of both functions is always $[0, 1]$. Another way to describe survivability is by the relative remaining

performance after a failure, i.e., M_{ij}^h/M_{ij}. Let $S_D(d)$ be the average remaining relative performance between node pairs that are at a DNSP of d from fault, then

$$S_D(d) = 1 - L_D(d). \tag{43}$$

Equation (43) can be verified using the definition in Equation (39):

$$\begin{aligned} L_D(d) &= \frac{1}{n}\sum_{h=1}^{n}\frac{1}{|I_d^h|}\sum_{i,j\in I_d^h}\frac{M_{ij}-M_{ij}^h}{M_{ij}} \\ &= \frac{1}{n}\sum_{h=1}^{n}\frac{1}{|I_d^h|}\sum_{i,j\in I_d^h}\left\{1-\frac{M_{ij}^h}{M_{ij}}\right\} \\ &= 1-\frac{1}{n}\sum_{h=1}^{n}\frac{1}{|I_d^h|}\sum_{i,j\in I_d^h}\frac{M_{ij}^h}{M_{ij}} = 1 - S_D(d). \end{aligned} \tag{44}$$

Another factor that impacts the amount of loss in flow and connectivity between nodes i and j due to failure of node h is the SPL between i and j. Intuitively, when the nodes are far away from each other, it is expected that they will have more alternative paths connecting them that are of equal or nearly equal length. Hence, the flow and connectivity of the two nodes is less likely to be affected significantly by failure of a single node.

Similarly to Equation (40), the expected loss between node pairs whose SPL is l due to a fault at node h is

$$L_L(l,h) = \frac{1}{|I_l|}\sum_{i,j\in I_l}\partial M_{ij}^h, \tag{45}$$

where $I_l = \{(i,j) \in V^2 \text{ such that } l_{ij} = l\}$.

Therefore, assuming all node failures are equally probable and that there is only a single node failure, the ARL between node pairs as a function of the SPL is

$$L_L(l) = \frac{1}{n}\sum_{h=1}^{n} L(l,h). \tag{46}$$

The ARL between node pairs of similar SPL and at same DNSP from fault is

$$L_{DL}(d,l) = \frac{1}{n}\sum_{h=1}^{n}\frac{1}{|I_{dl}^{h}|}\sum_{i,j\in I_{dl}^{h}}\partial M_{ij}^{h}, \tag{47}$$

where $I_{dl}^{h} = \{(i,j) \in V^2 : l_{ij} = l \text{ and } d_{ij}^{h} = d\}$.

The corresponding loss functions for the two performance measures, max-flow and connectivity, are L^F and L^C respectively. For large networks, it is impractical to compute max-flow for all $n(n-1)(n-2)$ fault node and node pair combinations. Therefore, a statistical mean of the max-flow and connectivity measures computed on a randomly selected origin-destination pairs and fault nodes is used to determine the mean loss at a given distance.

4.2.1 Numerical Simulations

The proposed approach proceeds as follows: assuming there is a random node failure, three subsets of equal size are generated from the node set of the network randomly. Nodes in the first set are designated as source nodes, the second set as destination nodes and the third set as fault nodes. The max-flow and connectivity between each node pair, (i,j), are computed before any fault and also after each case of failure, h.

Three networks will be used in this simulation: the highway networks of North Carolina, South Carolina and Tennessee. All the networks used are undirected and connected. Six network level indices are used to describe the networks as shown in Table 4.1. Connectivity in Table 4.1 is obtained by taking the average of all node pair connectivities. Connectivity, average clustering, and average degree measure redundancy and availability of alternative paths in the network. Hence higher value of these measures is desirable to ensure survivability.

Table 4.1. Network level indices of networks used for experiment

	$\overline{\deg ree}$	$\overline{clustering}$	$\overline{l}$	$\overline{C}$	D	n
NC	3.24	0.13	24.21	2.71	70	1534
SC	3.20	0.10	17.42	2.56	44	849
TN	3.16	0.09	22.86	2.61	68	1024

The contour plot shown in Figure 4.3 indicates the DNSP of node pairs from a fault is dependent on their SPL. The most common node-pair to fault node combinations have medium SPL and have small DNSP to fault nodes.

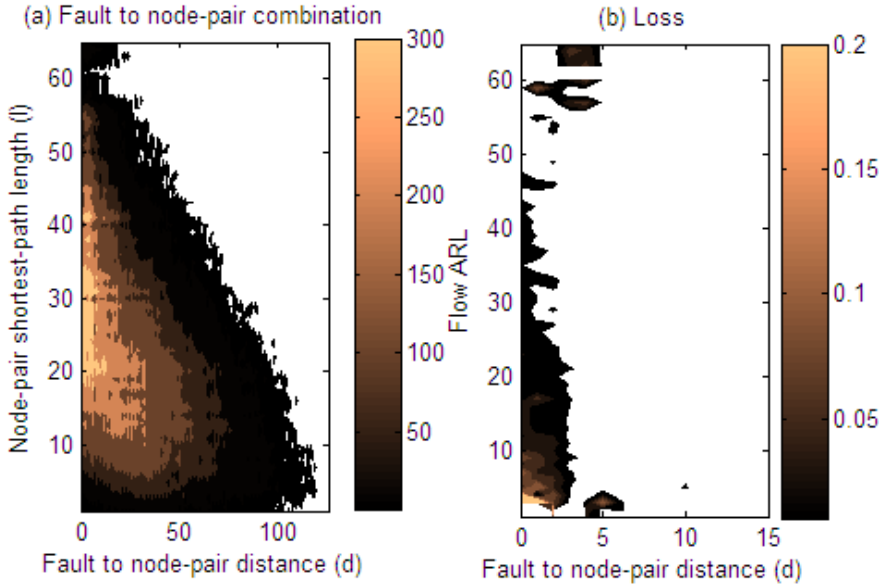

Figure 4.3. DNSP vs. SPL contour plots (a) the number of node triples (i, j, h) detected, (b) the average relative flow loss.

4.3 Relationship between Loss and DNSP

The dependence of the loss function on the DNSP between node pair and fault is described using Equation (41). Figure 4.4 illustrates this dependence graphically from experimental results on the NC and SC highway networks. As shown in Figure 4.4 the logarithm of max-flow loss, L_D^F can be fitted to a linear function of DNSP. Therefore, the

flow loss functions is approximated as an exponential function, i.e.,

$$L_D^F(d) = A_F e^{-r_F d}. \tag{48}$$

Let $d_F = 1/r_F$ be defined as the flow distance constant of the network, then Equation (48) suggests that there is a 63% decrease in the average flow loss between node pairs as their distance from the fault node increases by d_F.

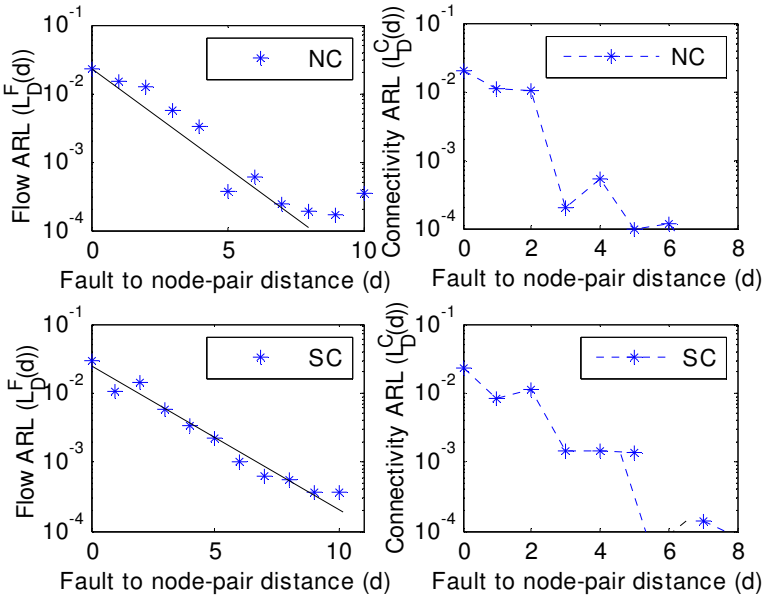

Figure 4.4. Logarithms of average loss of measure of interest in node pairs

The loss in connectivity does not show a clear pattern as the distance increases. But it decreases sharply at a distance d_c, defined here as a cutoff distance. The parameters of the loss functions can be used to define various survivability measures. The magnitude of the losses $A_F = L_D^F(0)$ and $A_C = L_D^C(0)$ at $d = 0$, i.e., when the fault node is in the shortest path of the node pairs, naturally represent the maximum loss incurred on node pairs. The rate of decay of the flow loss, r_F, shows how quickly the impact of fault decreases as one moves away from the fault node. And the connectivity loss cutoff distance, d_c, where

$L_D^F(d)$ is negligible for $d > d_c$, shows the reach of the impact.

Using max-flow and connectivity as performance measures, the survivability of a network can be described by four indices: (a) the flow distance constant, d_F; (b) the connectivity cutoff distance, d_c; (c) the flow loss at $d = 0$, A_F; and (d) the connectivity loss at $d = 0$, A_C. Naturally, a survivable network exhibits smaller values for all the four indices. The cutoff distance is obtained by setting 0.01% as a threshold for negligible loss, i.e., losses less than 10^{-4} are considered as no loss. The results of the experiments conducted on NC, SC and TN highway networks is summarized in Table 4.2.

Table 4.2. Coefficients of loss versus distance functions

Networks	A_F	d_F	A_C	$d_c(L^C \geq 10^{-4})$
NC	0.027	1.72	0.028	6
SC	0.041	1.72	0.039	7
TN	0.037	2.08	0.042	6

All the survivability measures indicate that the network in NC has the best survivability of the three. This can be explained by the fact that NC has the highest average clustering and marginally highest average connectivity and average degree of the three. All of which are indication of tightly and highly connected networks, which usually implies a better chance of survivability.

4.4 Impact of SPL on Performance

The results in the previous section are obtained by assuming that there is no distinction between node pairs with different SPL as far as the expected loss is concerned. But the contour in Figure 4.3b shows that the loss is dependent on the SPL. Although they have the same mean, SPL and DNSP have different distributions as shown in Figures 4.5a and 4.5b. If the range of the SPL is trisected into R_1 (short), R_2 (medium), and R_3 (far), the

loss functions decay differently for each group as shown in Figures 4.5c and 4.5d..

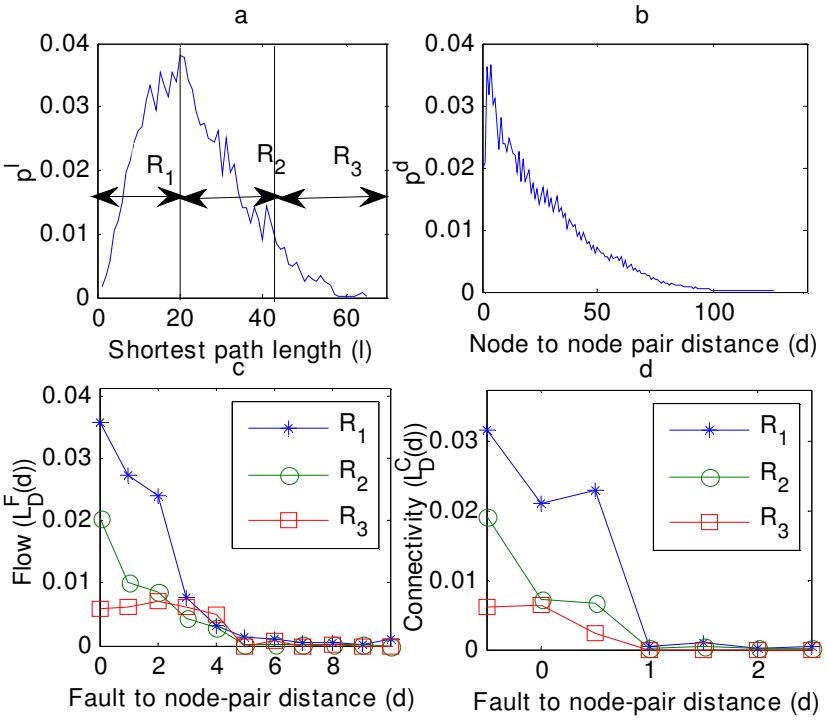

Figure 4.5. (a) Probability density function (pdf) of the SPL in NC. (b) pdf of DNSP. (c,d) Expected flow and connectivity loss as a function of DNSP grouped separately.

Since the reach of the impact is limited to the neighborhood of the fault, only the node to node-pair combinations whose DNSP is lower than the cutoff distance should be considered while evaluating the dependence of the loss on the SPL. Therefore, Equation (46) is revised to

$$L_L(l) = \frac{1}{n}\sum_{h=1}^{n}\frac{1}{|I_l^h|}\sum_{i,j\in I_l^h}\Delta M_{ij}^h, \tag{49}$$

where $I_l^h = \{(i,j) \in V^2$ such that $l_{ij} = l$ and $d_{ij}^h \leq d_c\}$

Figure 4.6 shows a clear trend of decreasing loss with an increase in SPL. There are two anomalies in this trend. The first is when the SPL is equal to 1. The SPL between nodes is equal to one only when the two nodes are directly connected, i.e., no intermediary node.

Therefore, from the definition of DNSP, Equation (33), there is no node h such that d_{ij}^h is zero. But, as shown in Figure 4.4, loss is significantly higher when the distance between fault node and node pair is zero. This suggests that directly linked node pairs are spared the worst loss, and hence the lower values shown in Figure 4.6.

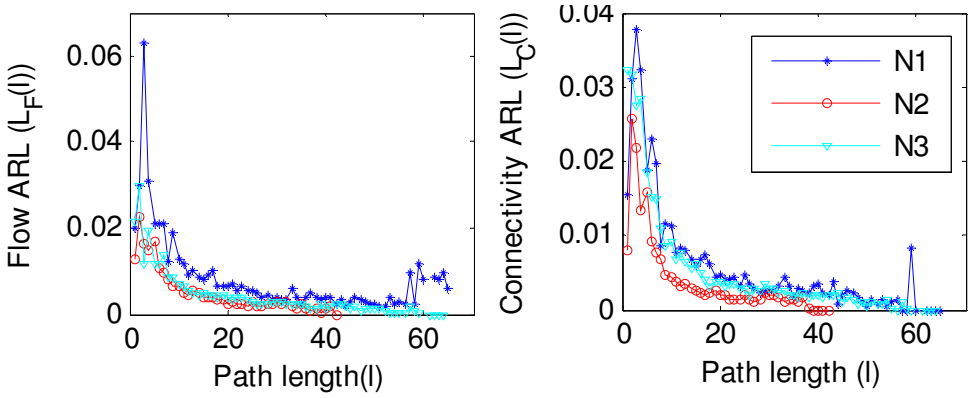

Figure 4.6. Average loss vs. SLP computed using eq. (49).

The second anomaly applies to node pairs that are very far apart. These node pairs are located in diametrically opposite fringes of the network, and have usually a very low degree. Therefore, they are heavily dependent on their few neighbors for their interaction with each other. This makes the node pairs vulnerable to single node failures.

According to Figure 4.5, the loss in R_1 and R_2 decays steadily, while it decays much slower in R_3 where the anomaly occurred in Figure 4.6. R_3 and node pairs whose SPL is 1 are removed when computing the log-log relation between SPL and loss shown in Figure 4.7. Therefore, both max-flow and connectivity losses can be approximated as

$$L_L^F(l) = K_F l^{-\gamma_F}, \tag{50}$$

$$L_L^C(l) = K_C l^{-\gamma_C}. \tag{51}$$

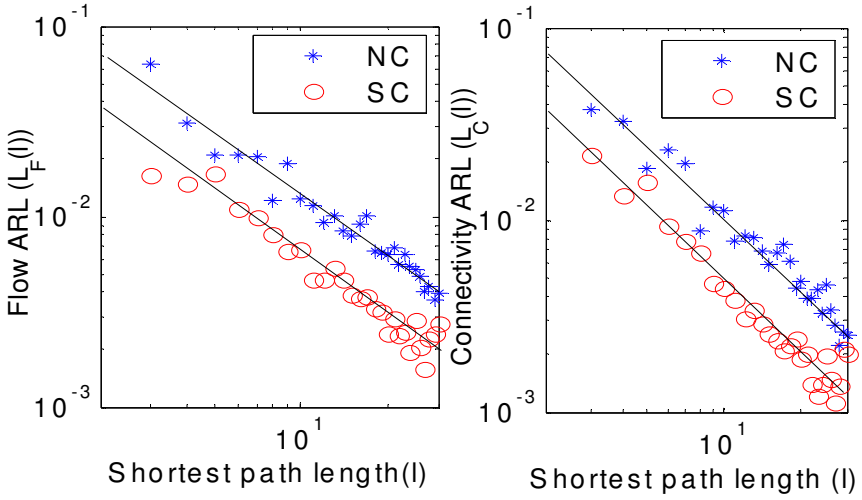

Figure 4.7. Log-log plot of SPL vs. loss after removing the anomalies in figure 4.6.

The function parameters of L_L^F and L_L^C for NC, SC and TN highway networks are summarized in Table 4.3.

Table 4.3. ARL vs SPL decay constants

Networks	K_F	γ_F	K_C	γ_C
NC	0.16	0.93	0.19	1.09
SC	0.08	0.76	0.11	0.99
TN	0.07	0.68	0.12	0.87

Table 4.3 shows that the NC network has the highest K_F and K_C but also has the sharpest decay rates. This suggests the L_L will be very small for l around the characteristic path length where most node-pairs are located. This is also reflected in the better survivability indices shown in Table 4.1.

CHAPTER 5

GEOGRAPHIC-BASED NETWORK COARSENING

Large and complex networks are computationally expensive to visualize, and predicting their responses to faults and attacks is even more challenging. In addition, in infrastructure networks interactions between different sections of the network are at times more important than node-to-node interactions. For instance, the flow between two road intersections is much less important than the flow between two parts of a city or even between two cities.

To address these two problems in spatial networks, i.e., the need for neighborhood-based analysis, and to avoid computational explosion, this work proposes a way of clustering the network's nodes according to their geographic locations and their connectivity constraint. Each cluster represents a geographic neighborhood, and the interactions of the neighborhoods form the coarsened network.

To coarsen a network there are mainly three issues that need to be addressed. (a) The first is how to divide the node set into different groups; (b) The construction of the topology of the coarsened network needs to be addressed; and (c) The node and link parameters of the coarsened network needs to be determined.

5.1 Algorithms for Partitioning the node set

The first mechanism involves partitioning the node set of a network into several groups. In spatial networks, these groups represent geographic regions if they are contiguous. A region can be defined using graph properties, e.g., a subgraph g with diameter d, or geometric properties, e.g., a circular area of radius r in the network layout [88]. In this

work, a region is defined as a mega-node formed by collapsing all the nodes that share same geographic neighborhood as determined by a cluster-seeking or graph-partitioning algorithm. A region can also be viewed as a subset of the node set of the network, $R \subseteq V$, closest to a cluster center. The average size of a node in the coarsened node, i.e., the number of nodes in the original network represented by a single node in the coarsened network defines the coarsening ratio. The size of a region can be defined as some characteristic distance of the graph or the level of coarsening desired.

This work proposes two ways of partitioning the node space. The first method, clustering by node location, assumes that the geographic coordinates of nodes are known. The algorithm groups the nodes into clusters using standard clustering techniques while satisfying connectivity constraints. The second method is a minimum weight partitioning. This method finds a proximity weight between two nodes and employ a partitioning algorithm that breaks the network into the desired number of components while minimizing the total cost of edges cut.

5.1.1 Clustering by node location

In a typical k-mean clustering algorithm, a point is assigned to a cluster if the centroid of the cluster is the closest to the point than all other centroids. Here, a point is assigned to a cluster whose centroid is closest to the point among all clusters that contain at least one node that has a direct link to the node represented by the point. This algorithm returns the clustered set and takes as an input the number of clusters desired ($k = |V^R|$), adjacency matrix of the network (A), and the locations of the nodes (P), where $P_i = (x_i, y_i)$ is the geographic coordinate where the node i is located. This algorithm has a worst case

running time of O($|V||V^R|N$), where N is the maximum number of iterations. The algorithm is similar to k-mean except the step where a point is assigned to a cluster and a cleaning up step at the end. A typical iteration of the modified k-mean algorithm is shown in Figure 5.1.

```
Input: Adjacency matrix (A), node locations (P), number of clusters (k)
Output: cluster indices of each node (B)
B = ModifedKMean (A, P, k)
...
for each node
     select closest cluster with the most nodes connected to current node
     assign node to cluster and modify centroid
end for
...
for each cluster
    while cluster is not connected
       move disconnected node to nearest cluster with most nodes connected to it
    end while
 end for
```

Figure 5.1. A modified k-mean algorithm.

The second *for* loop in the modified k-mean algorithm is a cleaning-up step that is required to make sure that all the clusters are contiguous.

5.1.2 Minimum-weight partitioning

Regions, or neighborhoods in a network can be identified using a grouping approach that minimizes a certain type of proximity weight of inter-regional links. Hence, in spatial networks the notion of physical proximity must be defined. This is achieved by transforming the distance and the adjacency matrix of the network into a proximity weight matrix. In this work, the proximity weight between two nodes is defined as the weighted average of the physical distance between the given nodes and the size of their shared neighbor-

hood. Let $g_d(i,j)$ be the geographic distance between nodes i and j, then a distance-based weight, W_D, can be defined as

$$W_D(i,j) = 1 - \frac{g_d(i,j) - \min(g_d)}{\max(g_d) - \min(g_d)}. \tag{52}$$

Thus, $W_D \in [0,1]$. The extreme cases of 0 and 1 occur at the longest and shortest links respectively.

The second proximity measure assumes that nodes in the same geographic neighborhood tend to have more common neighbors than nodes of different neighborhoods. In this work, two types of neighbors are considered: (a) immediate neighbors and (b) secondary neighbors. Let the immediate neighbors of node i be $I_i = \{h \in V : A(i,h) = 1\}$and the secondary neighbors of node i, be $I_i^2 = \{h \in V \backslash I_i \text{ and } k \in I_i : A(k,h) = 1\} \backslash \{i\}$. Then the number of first common neighbors between i and j, F, and the number of second common neighbors, S, are

$$F(i,j) = |I_i \cap I_j|, \tag{53}$$

$$S(i,j) = |I_i \cap I_j^2| + |I_i^2 \cap I_j^2| + |I_i^2 \cap I_j|. \tag{54}$$

Equations (52)-(54) can be combined to define an overall weight of link (i,j). Therefore, the proximity weight between nodes i and j is

$$W(i,j) = \begin{cases} \omega_d W_D(i,j) + \omega_F F(i,j) + \omega_S S(i,j) & \text{if } A(i,j) = 1 \\ 0 & \text{if } A(i,j) = 0 \end{cases} \tag{55}$$

where $\omega_d + \omega_F + \omega_S = 1$. An example of common neighborhood counting is shown in Figure 5.2.

Once the weights of all the links are determined, any weight-based partitioning algorithm can be used to partition the node set. In this work, the *gpmetis* program of

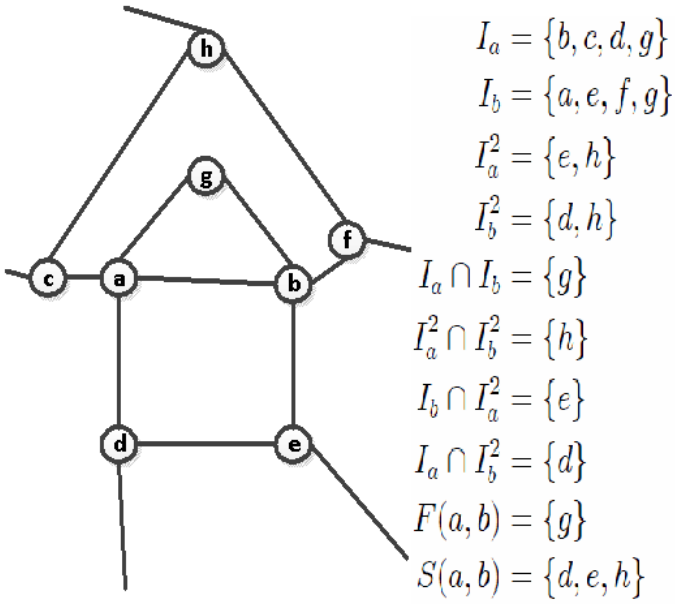

Figure 5.2. Computation of proximity weight due to common neighborhood.

METIS multilevel graph partitioning software [89] is used to obtain high quality partitions. The *gpmetis* is set to return contiguous partitions using a k-way cut, where k is the number of partitions desired. It tries to minimize the imbalance between partition size and the total weight of edges cut. The partitions returned from *gpmetis* are used to build the adjacency matrix of the region network as shown in Equation (56).

5.2 Construction of Coarsened Network

The second issue in network coarsening addresses how the network of regions is represented as a graph. As a compression strategy, coarsening will result in a loss of information. What information to lose and what information to keep depends on the purpose of the coarsening. There are several ways of representing the links of the coarsened network. These representations are desired to reflect application specific properties. Figure 5.3 shows three types of representation, assuming that the original graph is simple graph,

i.e., that the graph contains neither loops nor parallel edges.

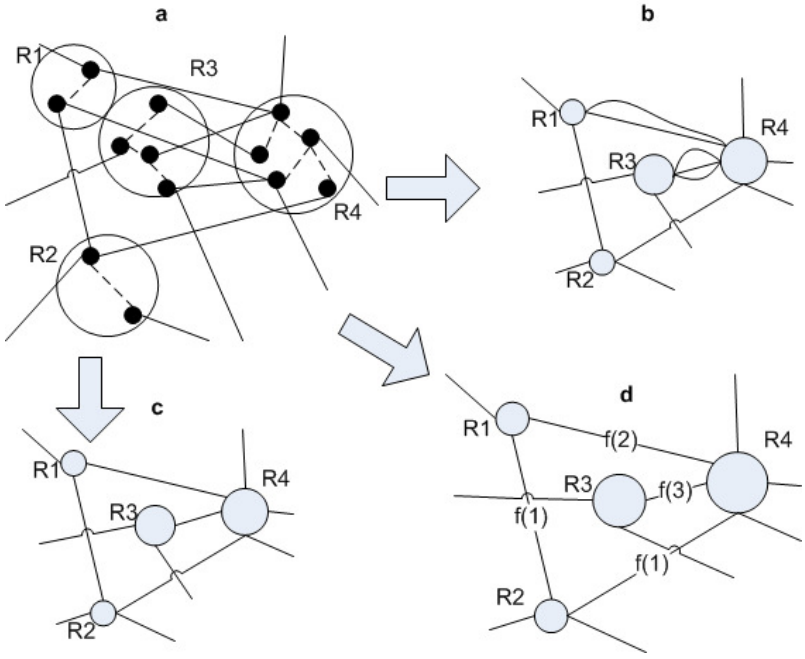

Figure 5.3. Coarsened network representations. (a) Original network. (b) Multi-graph. (c) Simple graph. (d) Weighted graph.

Each of the three representations shown in Figure 5.3 has its own merit. In the multi-graph representation the number of links incident on the same two regions is preserved, while in the simple graph only the availability of connection is shown. The weighted graph, with weight assigning function f of the number of links between regions, can take several interpretations; for instance, time-delay or capacity information between regions. Hence the higher the number of links the lesser the delay and the higher the capacity.

Figure 5.3 deals with the different ways of constructing inter-region links. Another important issue is how a region is represented in the coarsened network. The simplest case is to represent the region as a single node as shown in Figure 5.3. A region can also be represented as a star graph as shown in Figure 5.4. For instance, region R_a is

represented by a central node A, which is made up of all intermediate nodes of the region, and peripheral nodes ab, ac and ad, which are made up of nodes bordering regions R_b, R_c and R_d respectively. The intra-region links can also be defined similarly to the inter-region links to reflect traffic delay, capacity, number of links, etc.

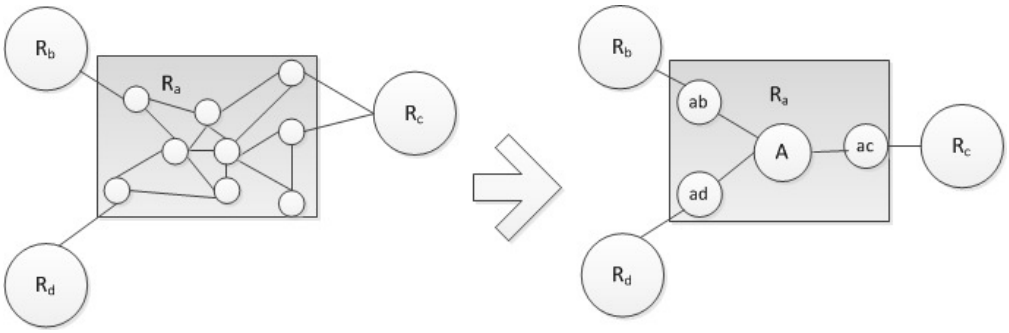

Figure 5.4. Representing a region with multiple nodes.

In addition to the construction mechanisms discussed so far, various topologies can be proposed depending on the motive for coarsening. For instance, loops can be used to show the number of intra-region links, nodes of the coarsened network can be weighted to show their size. A multi-node representation of a region can be used to show a regions shape if, for instance, it is unproportionally wide.

5.2.1 Adjacency Matrix

A graph representation of a network is composed of nodes and links. Nodes and links can be described by several numbers that are used to describe the network topology, link capacities, geometric distances and etc. These descriptors of nodes and links of the coarsened network depend on the construction strategy employed.

Let a region be represented by a single node and the number of links originating from region i, R_i, and terminating in region j, R_j be $L(i, j)$. Let A be the adjacency

matrix of the original graph, where $A(i,j)$ is the number of links from node i to node j. Then, the adjacency matrices of the simple, A^{R_s}, and multigraph, A^{R_m}, representation of the coarsened network are,

$$L(i,j) = \sum_{k \in R_i} \sum_{h \in R_j} A(k,h) \tag{56}$$

$$A^{R_s}(i,j) = \begin{cases} 0 & \text{if } L(i,j) = 0 \\ 1 & \text{otherwise} \end{cases} \tag{57}$$

$$A^{R_m}(i,j) = L(i,j) \tag{58}$$

For instance, the single node representation of the network shown in Figure 5.4 is

$$A^{R_s} = \begin{array}{c|cccc} & R_a & R_b & R_c & R_d \\ R_a & 0 & 1 & 1 & 1 \\ R_b & 1 & 0 & 0 & 0 \\ R_c & 1 & 0 & 0 & 0 \\ R_d & 1 & 0 & 0 & 0 \end{array} \quad \text{and} \quad A^{R_m} = \begin{array}{c|cccc} & R_a & R_b & R_c & R_d \\ R_a & 0 & 1 & 2 & 1 \\ R_b & 1 & 0 & 0 & 0 \\ R_c & 2 & 0 & 0 & 0 \\ R_d & 1 & 0 & 0 & 0 \end{array} .$$

The multi-node representation of regions can be achieved by expanding Equations (56)-(58).

5.2.2 Link and node weights

In the multigraph representation, the capacity of inter-region links will remain the same. But in the simple graph representation, the capacity of the link between two regions is an aggregate of all the capacities between the given regions. Let $F(i,j)$ be the capacity of a link between nodes i and j, then the capacity of a link connecting two regions, R_i and R_j, in a simple graph is formulated as

$$F^R(i,j) = \sum_{k \in R_i} \sum_{h \in R_j} F(k,h) \tag{59}$$

The distance between the regions can be defined as the distance of the shortest arc connecting any node $k \in R_i$ and any node $h \in R_j$, the distance between the centers of the regions, the average distance between two nodes belonging to the different regions, etc.

For instance,

$$D^R(i,j) = \frac{1}{|R_i||R_j|} \sum_{k \in R_i} \sum_{h \in R_j} D(k,h) \tag{60}$$

Node weights are associated with nodes to identify a region's internal characteristics (since nodes in the coarsened network represent regions in both single node and multi-node representations). The weights associated with a node can be used to describe the corresponding network property of the subgraph represented as a region, such as number of nodes/links or other network parameters such as the diameter and density.

5.3 Effect of Coarsening on Network Parameters

Figures 5.5 - 5.7 show how some of the network important measures change as the level of coarsening increases in the NC and SC highway networks. The network properties considered here are the characteristic-path length, average clustering, and average connectivity. In this work, connectivity between a given node pair i, j is defined as the number of edge-disjoint paths from i to j. The average connectivity is an average over all node-pairs.

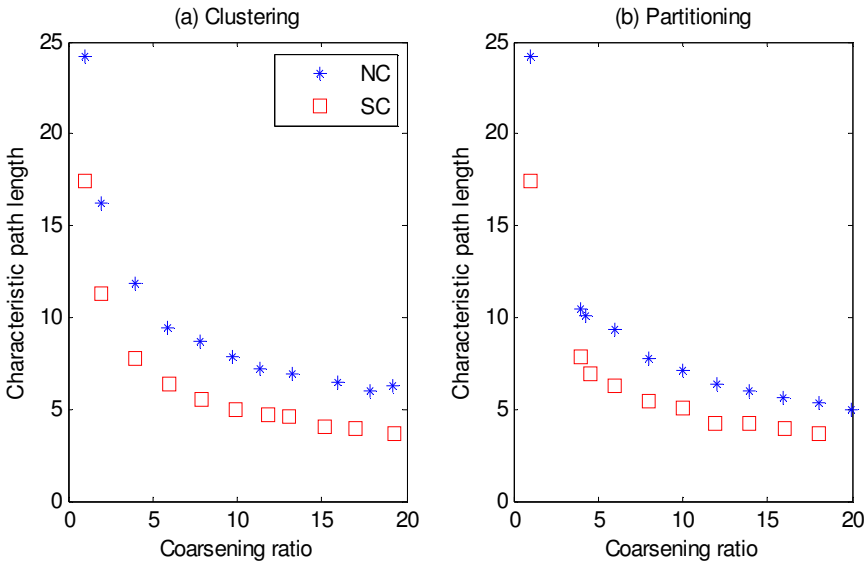

Figure 5.5. Characteristic path length vs. coarsening ratio.

The characteristic path length of a network is the number of average hops; therefore a simple graph representation is used. A simple graph representation is also used to find the average clustering in the network, since multiple edges between two given nodes can make a nodes clustering exceed the clustering coefficient $[0, 1]$. On the other hand, the number of disjoint paths between two regions depends on the number of links between the regions. Hence, a multigraph representation is more fitting to analyze the impact of coarsening on network's average connectivity.

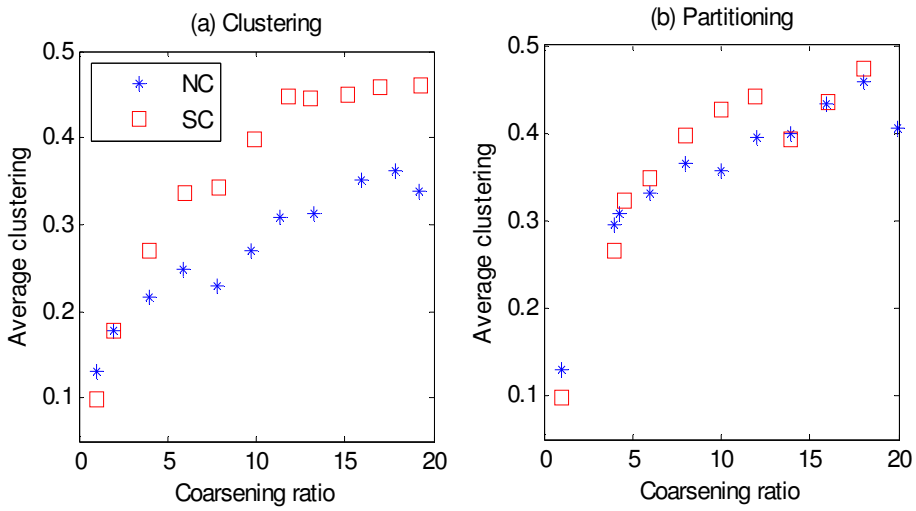

Figure 5.6. Network clustering vs. coarsening ratio.

From Figure 5.7, it can be seen than there is no pattern between coarsening ratio and average connectivity when minimum weight partitioning algorithm is used to partition the node set. This is because minimizing the total weight of edges cut also minimizes the number of edges cut. Hence, the increase in inter-region links is not proportional to the increase of the coarsening level. This hinders the average connectivity from increasing proportionally with the coarsening ratio.

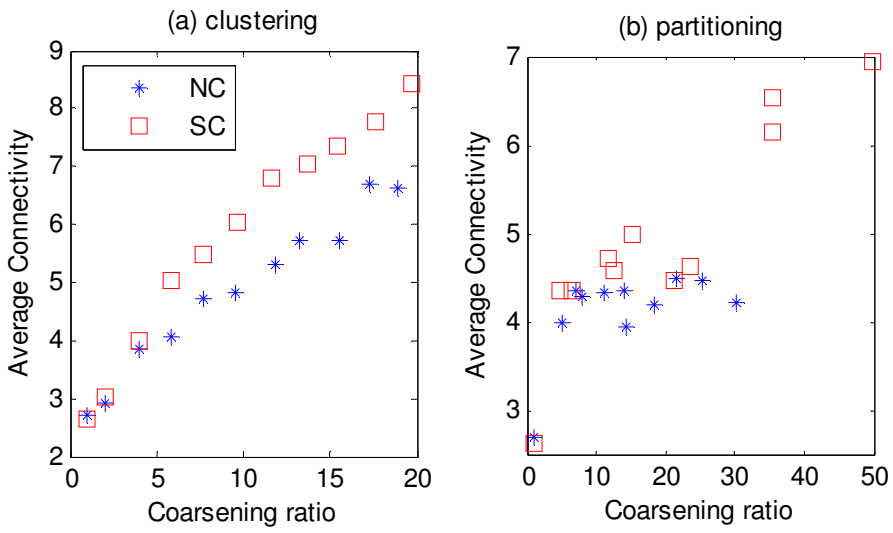

Figure 5.7. Network connectivity vs. coarsening ratio.

5.4 Application of Coarsening: Complexity Reduction

5.4.1 Computational Complexity reduction

To find the characteristic-path length of a network, the easiest way is to invoke Dijkstra's algorithm $|V|$ times and then take the average of the computed path lengths. This has a complexity of $O(|V|^2 \log |V| + |V||E|)$. For large networks, this algorithm will require large resources.

This work proposes a way of partitioning the network into smaller sub-networks and finding desired network parameters on all the sub-networks. The parameters are then merged using probabilistic models to estimate the original network measures as shown in Figure 5.8. Methods to find the characteristic-path length and average clustering of a network are developed. From here onwards distance between two nodes refers to the length of the shortest path between them.

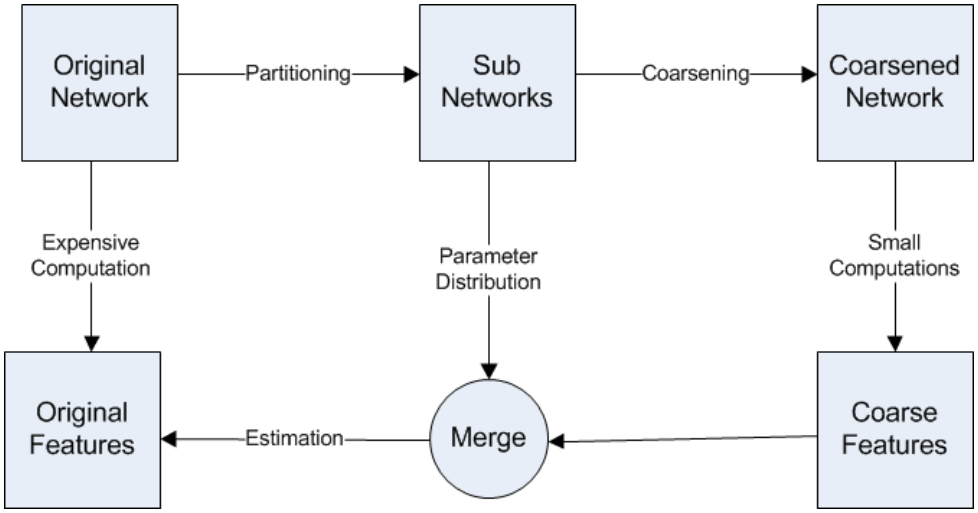

Figure 5.8. Basic structure of coarsening based parameter estimation model.

5.4.1.1 Characteristic-path length

The shortest path between R_i and R_j in a coarsened network has a form

$$R_{i_1} - R_{i_2} - R_{i_3} - ... - R_{i_{m-1}} - R_{i_m}$$

where m is the number of regions in the path, $R_{i_1} = R_i$ and $R_{i_m} = R_j$. Let the distance D_{hk} between two adjacent regions R_h and R_k be the average length of links connecting the boundaries of R_h and R_k. Let W_k be the width of R_k measured as the distance from one boundary to another in R_k. Then the distance between R_i and R_j is formulated as

$$|R_i - R_j| = \sum_{k=2}^{m-1} W_{i_k} + \sum_{k=1}^{m-1} D_{i_k,i_{k+1}} \tag{61}$$

The distance between two nodes in different regions, for instance, nodes p_i in R_i and p_j in R_j in Figure 5.9, can be approximated as the sum of the distances between the nodes to their respective boundaries and the distance between R_i and R_j, $|R_i - R_j|$. Let B_{hk} denote the boundary between R_h and R_k, i.e., the set of nodes in R_h that are connected

to nodes in R_k. Then the distance between p_i and p_j is

$$|p_i - p_j| = |p_i - B_{i_1,i_2}| + |p_j - B_{i_m,i_{m-1}}| + |R_i - R_j| \tag{62}$$

Equation (62) is applicable only if $R_i \neq R_j$.

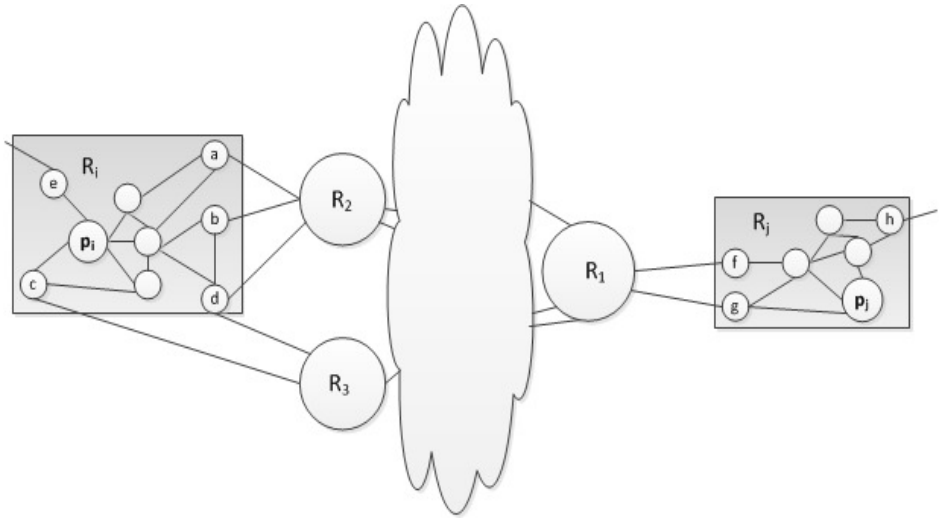

Figure 5.9. Illustration of graph complexity reduction.

The characteristic-path length of a network is the average distance between two nodes over all $(i, j) \in V^2/i \neq j$. The expected distance between any two nodes in the network that belong to different regions is $L'_N = E[|p_i - p_j|]$, and it can be found from Equations (61) and (62) as the expected value of $|p_i - p_j|$.

$$L'_N = E\left[|p_i - B_{i_1,i_2}| + |p_j - B_{i_m,i_{m-1}}| + \sum_{k=2}^{m-1} W_{i_k} + \sum_{k=1}^{m-1} D_{i_k,i_{k+1}}\right] \tag{63}$$

$$= 2E[|p_i - B_{i_1,i_2}|] + (\bar{m} - 2)\bar{W} + (\bar{m} - 1)\bar{D} \tag{64}$$

where $\bar{m}$ is the average number of hops between any two regions of the coarsened network. Since $\bar{m}$ and $\bar{D}$ can be found while constructing the coarsened network, they will not be discussed any further. To find $E[|p_i - B_{i_1,i_2}|]$ and $\bar{W}$, let a random variable $X_h^{i_1,i_2}$ be defined as the distance between a random node in R_{i_1} and node h in B_{i_1,i_2}. Then, the

shortest distance between a random node in R_{i_1}, and any of the nodes in B_{i_1,i_2}, is

$$Y^{i_1,i_2} = \min\{X_1^{i_1,i_2}, X_2^{i_1,i_2}, ..., X_k^{i_1,i_2}\} \tag{65}$$

where k is the cardinality of the set B_{i_1,i_2}. For the rest of this section the superscript i_1, i_2 will be dropped. The probability that the distance is less than z, where z is a dummy variable, is thus given by;

$$\Pr(Y \leq z) = \Pr(\min\{X_1, X_2, ..., X_k\} > z\}') \tag{66}$$

$$= 1 - \Pr\left(\cap_{h=1}^{k}\{X_h > z\}\right) \tag{67}$$

Assuming X_h is independently and identically distributed (iid), Equation (67) can be simplified as

$$\Pr(Y \leq z) = 1 - [\Pr(X_h > z)]^k = 1 - [1 - F_{X_h}(z)]^k \tag{68}$$

where $F_{X_h}(z)$ is the cumulative distribution function of X_h. Therefore, the average distance between a random node in a region and the region's boundary is

$$E[Y] = \int_{-\infty}^{\infty} z f_Y(z) dz = k \int_0^{D_R} z F'_{X_h}(z)[1 - F_{X_h}(z)]^{k-1} dz \tag{69}$$

where D_R is the graph diameter of the region.

The width of a region (the average distance between two boundary sets of a region) can be obtained following similar procedure. Let the two boundaries of R_{i_2} with R_{i_1} and R_{i_3} be B_{i_2,i_1} and B_{i_2,i_3} respectively. Let $X_{hh'}^{i_1,i_2,i_3}$ be the distance between nodes h in B_{i_2,i_1} and h' in B_{i_2,i_3}. Then, assuming $X_{hh'}^{i_1,i_2,i_3}$ is iid, the shortest distance between the two boundary sets is

$$W^{i_1,i_2,i_3} = \min\{X_{11}^{i_1,i_2,i_3}, X_{12}^{i_1,i_2,i_3}, ..., X_{1m}^{i_1,i_2,i_3}, X_{21}^{i_1,i_2,i_3}, X_{22}^{i_1,i_2,i_3}, ..., X_{ks}^{i_1,i_2,i_3}\} \tag{70}$$

where k is the cardinality of set B_{i_2,i_1} and s is the cardinality of set B_{i_2,i_3}. Thus, the average

distance between two boundary sets, i.e., the average width of a region, is

$$E[W] = \int_{-\infty}^{\infty} z f_W(z) dz = sk \int_0^{D_C} z F'_{X_{hh'}}(z)[1 - F_{X_{hh'}}(z)]^{ks-1} dz \tag{71}$$

Therefore, the average distance between nodes of different region is given by $L'_N = 2\bar{Y} + (m-2)\bar{W} + (m-1)\bar{D}$.

Figure 5.10 shows the distribution of X, Y, and W in the NC highway network with 1:20 coarsening ratio. As expected, the mean of the distributions moves to the left as the number of nodes increases.

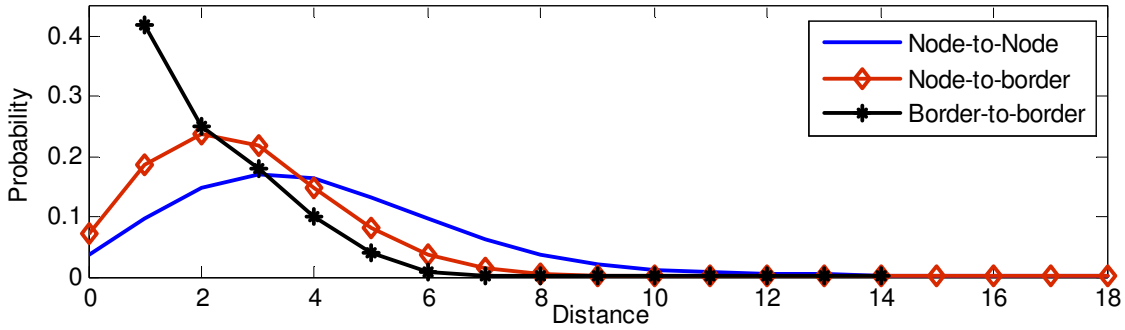

Figure 5.10. Average distributions of shortest path inside regions.

Equation (62) finds the average distance between two nodes belonging to different regions. To find the overall average distance of a network, a weighted average of the distances between the intra-region node-pairs and inter-region node-pairs is used. Let U be a random variable representing the distance between two nodes in the same region. Let ζ be the probability that the two nodes in a given node pair belong to same region. Then the expected distance between the two nodes can be approximated by

$$L_N \approx E[U]\zeta + L'_N(1 - \zeta) \tag{72}$$

where $E[U]$ is the average intra-region distance. Let r be the average number of nodes in a

region, then

$$\zeta = \frac{\frac{|V|}{r} r(r-1)}{|V|(|V|-1)} = \frac{r-1}{|V|-1} \tag{73}$$

Thus, the overall characteristic-path length of the original network is

$$L_N \approx \frac{L'_N(|V|-r) + E[U](r-1)}{(|V|-1)} \tag{74}$$

At the extreme case, where each node is a region, i.e., $r = 1$, $L_N = L'_N$ and since there is only one node in the region $X = W = Y = 0$. Therefore, $L_N = \acute{L}'_N = (\bar{m}-1)\bar{D}$. Which is the characteristic path length of the coarsened network. In addition, if the entire network is a single region, i.e., $r = |V|$ and $\bar{D} = 0$, then $L_N = E[U]$, which is the characteristic path length of the region. This shows that the approximation of average distance using Equation (74) gives exact solutions at the two extreme cases of coarsening.

5.4.1.2 Average Clustering

Graph clustering of a node h in a given network is

$$c_h = \frac{\Delta_h}{\delta_h(\delta_h - 1)}, \tag{75}$$

where Δ_h is the number of triangles node h belongs to and δ_h is its degree. Let $|V_i|$ and $\bar{C}_i$ be the number of nodes and the average graph clustering of R_i respectively. $\bar{C}_i$ is computed directly from the subnetwork represented by R_i. Assuming there are no links between nodes in different regions, the clustering of the network can be approximated as

$$\bar{C}^R = \frac{1}{|V|} \sum_{i=1}^{|V|/r} \bar{C}_i |V_i| \tag{76}$$

where $|V|/r$ is the number of regions in the network. Ignoring the inter-region links reduces the degrees of nodes, thus effectively increasing the network clustering. Let ν_h be the number of links connecting node h to nodes outside its region. Then assuming all the triangles containing node h are inside the region containing h, the graph clustering of h,

c_h^R, becomes

$$c_h^R = \frac{\Delta_h}{(\delta_h - v_h)(\delta_h - v_h - 1)}. \tag{77}$$

Equation (77) shows that a node's average clustering in the subnetwork of its region is greater than its clustering in the original network in Equation (75). Let ε_h be the node clustering discrepancy of node h between the region subnetwork and the original network, then

$$\varepsilon_h = \frac{\Delta_h}{(\delta_h - v_h)(\delta_h - v_h - 1)} - \frac{\Delta_h}{\delta_h(\delta_h - 1)} \tag{78}$$

$$= \frac{\Delta_h \nu_h (2\delta_h - v_h - 1)}{\delta_h(\delta_h - 1)(\delta_h - v_h)(\delta_h - v_h - 1)} \tag{79}$$

$$\varepsilon_h = c_h \beta_h \nu_h \tag{80}$$

where $\beta_h = \frac{2\delta_h - v_h - 1}{(\delta_h - v_h)(\delta_h - v_h - 1)}$ is the graph clustering overestimation factor per link for node h. Therefore the average network clustering is given by:

$$E\left[c_h\right] = E\left[c_h^R\right] - E\left[c_h \beta_h v_h\right]$$

$$\bar{C}_N = \bar{C}^R - \bar{C}_N \bar{\beta} \bar{\nu}$$

$$\bar{C}_N = \frac{\bar{C}^R}{1 + \bar{\beta}\bar{\nu}}$$

If the entire network is a single region, i.e., $r = |V|$ and $\nu_h = 0, \forall h \in V$, then $\varepsilon_h = 0$ and $c_h = c_h^R$.

If there are r nodes in a region on average, the time complexity of finding all shortest paths, using Dijkstra's algorithm, between nodes of the same region is

$$O(\frac{|V|}{r}(r^2 \log r + re)) = O(|V| r \log r + |V| e)$$

where $e = \frac{|E| r}{|V|}$ is the average number of links in a region. The probabilistic model used to merge these results from all the regions has a complexity of constant time. Thus the

complexity of this approach is $O(\frac{|V|}{r}(r^2 \log r + re))$ as opposed to $O(|V|^2 \log |V| + |V||E|)$. This significant complexity reduction comes at the cost of the coarsening procedure and from a less than exact answers. But, the cost of coarsening is offset by the fact that once the network is coarsened several other measures can be computed similarly. Table 5.1 shows the performance of the coarsening approach in predicting network level topological measures. A 1:10 coarsening ratio is used. In the table, the values in A are found from original network and the values in R are found using the above approximation strategies.

Table 5.1. Actual and estimated characteristic-path length and average clustering of highway networks

	C_N		L_N		$\|V\|$
State	A	R	A	R	
NC	0.13	0.14	24.21	24.71	1534
SC	0.10	0.10	17.42	18.48	849
TN	0.09	0.09	22.86	23.66	1024
VA	0.11	0.11	21.12	22.63	1220

The performance of location-based clustering depends on the initial condition and convergence of the clustering algorithm and hence the estimation of the above model is not unique. But, in general, clustering is much more predictable than characteristic path length and the estimation errors are positive. The worst error found in this experiment, 7.15%, corresponds to the VA highway characteristic path length.

5.4.2 Hierarchical Shortest-Path Routing

The previous section illustrates that a reasonable approximation of network statistics can be found without the actual network level computation. This is achieved by using the distributions in the coarsened network and the subnetworks represented by a region. This section considers another possible application of this coarsening approach. A

shortest-path routing problem. This is a fairly straightforward problem and has a polynomial solution. When the network size is very large, and finer details of the network are not available globally, the solution to this problem can be resource intensive. In addition, hierarchical routing methods are important in interconnected network systems, such as the internet.

Equations (61) and (62) represent a way of finding the shortest path between two nodes in a network using coarsening. Based on these equations, a simple greedy algorithm that finds the shortest path from a source node s in region R_i to a destination node d in region R_j is proposed as shown in Figure 5.11. For a path to be found via this approach the subnetworks represented by a region should be contiguous.

Figure 5.12 compares the shortest paths found by running shortest path algorithms directly on the original network (L) with those computed by using the above hierarchical algorithm (L') on the NC highway network. The distribution shows the cumulative distribution of L/L' over all node pairs of the network.

As shown earlier, there are $|V|(r-1)$ intra-region node-pairs for a coarsening ratio of r. The hierarchical algorithm is similar to the direct shortest path algorithm for these node pairs. Hence, the algorithm guarantees a minimum of $(r-1)/(|V|-1)$ exact routing. The CDF in Figure 5.12 is obtained for $r = 20$ and $|V| = 1534$. This guarantees a 1.24% exact routing. But the algorithm has achieved over 16% exact routing. In addition over 80% of the routes in the hierarchical routing are within 30% of the actual shortest path. On average, the hierarchical route is 23% longer than the shortest route.

Input: Partitions (B), coarsened network (R), source (s), destination (d)
Output: Shortest path between s and d (P)
P = HierarchicalRouting (B, R, s, d)

STEP 1: initialization
    $P_R \leftarrow$ find the shortest path in the coarsened network from $B(s)$ to $B(d)$
    current node, $c = s$
    current region, r = $B(s)$
    $regionCount = 2$
    next region, $h = P_R\ (regionCount)$
    Initial empty path, $P \leftarrow \{\}$
STEP2: find paths inside regions
    $P_{ck} \leftarrow$ find the path from c to the closest boundary node (k) from r and h
    $P \leftarrow \{P, P_{ck}\}$
    $P_{km} \leftarrow$ find the path from k to the closest boundary node (m) from h and r
    $P \leftarrow \{P, P_{km}\}$
    $regionCount$++
STEP3: stop condition
    if $h = B(d)$
        $P_{md} \leftarrow$ find shortest path from m to d
        return $P \leftarrow \{P, P_{md}\}$
    else
        $c=m$, $r=h$
        $h = P_R\ (regionCount)$
        go to step 2

Figure 5.11. Hierarchical routing algorithm.

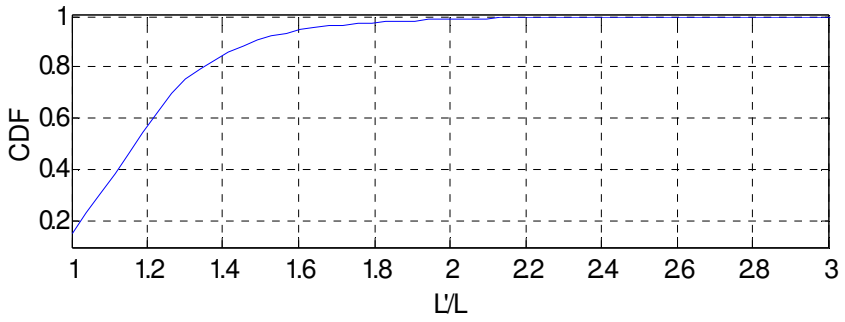

Figure 5.12. Comparison of shortest paths obtained hierarchically and directly.

CHAPTER 6

SMALL-WORLD PROPERTY OF ROAD NETWORKS

The study of small-world phenomena in networks often uses the rewiring of 1-D lattices to obtain highly ordered and globally compact networks. This approach has led to many important observations in social networks. In both grid plan and street hierarchy road network plans, a 2-D lattice can be considered as the backbone of the network. This chapter presents a study of network statistics and small world property of 2-D planar lattice network. The small-world property of a road network at different levels of view is also investigated. The different levels of view are defined by the coarsening ratio as defined in chapter 5.

6.1 Grid lattices and their network statistics

The simplest 2-D lattice is a grid with m horizontal and n vertical points. Thus, the graph of the lattice has $|V| = nm$ nodes. Lattice nodes are connected only to their lattice neighbors. In planar lattices, nodes are connected only to their physically immediate neighbors. In this work, the definition includes the node diagonally opposite to a given node as an immediate neighbor. This forms triangular cells. Figure 6.1 shows two types of 2-D planar lattices, rectangular and triangular.

6.1.1 Characteristic-path length of 2-D lattices

Let $d(i, j)$ be the number of edges in the shortest path between nodes i and j, then the characteristic-path length, mean of all the shortest paths between all node pairs is

$$L_N = \frac{1}{|V|(|V|-1)} \sum_{i=1}^{|V|} \sum_{j=1, j\neq i}^{|V|} d(i, j) \tag{81}$$

For a rectangular lattice the characteristic-path length can be formulated as a func-

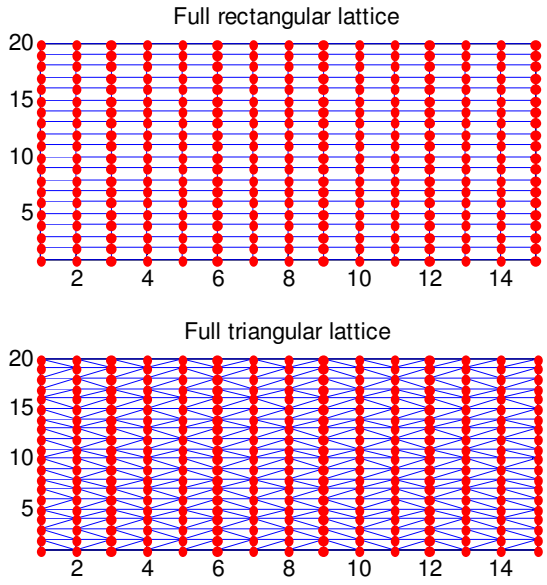

Figure 6.1. 2-dimensional rectangular and triangular planar lattices.

tion of the lattice dimensions. Let (i, j) be the node in the lower right corner of a lattice. The sum of all paths originating from (i, j) can be viewed as composed of three types of paths. These paths are: (a) paths to $i - 1$ nodes on the vertical line, (b) paths to $j - 1$ nodes on the horizontal line, and (c) paths to $ij - i - j + 1$ nodes that do not share any of (i, j)'s coordinates. The distance between nodes that do not share any of the coordinates with node (i, j) is equal to the distance of the nodes from node $(i - 1, j - 1)$ plus two.

Let $S(i, j)$ be the sum of the distances of nodes in the vertical and horizontal lines of node (i, j)

$$S(i,j) = \sum_{i'=1}^{i-1} i' + \sum_{j'=1}^{j-1} j' \tag{82}$$

and let $D_T(i,j)$ be the span of the minimum spanning tree with its root at node (i,j)

$$D_T(i,j) = S(i,j) + 2(ij - i - j + 1) + D_T(i-1, j-1) \tag{83}$$
$$= \frac{i(i-1)}{2} + \frac{j(j-1)}{2} + 2(ij - i - j + 1) + D_T(i-1, j-1)$$

$$D_T(0,x) = D_T(y,0) = 0 \tag{84}$$

Equations (82 - 84) can be solved using mathematical induction

$$D_T(i,j) = \frac{ij}{2}(i + j - 2) \tag{85}$$

From symmetry it can be shown that $D_T(1,j) = D_T(i,1) = D_T(1,1) = D_T(i,j)$.

A node (i,j) at any location in the lattice will have a maximum of four diagonal neighbors, thus dividing the lattice into four separate lattices as shown in Figure 6.2. The sum of the distance of all nodes from node (i,j) is computed similar to Equation (83), except that in this case there are four diagonal neighbors; i.e.,

$$D_T(i,j) = S(i,j) + 2(mn - m - n + 1) + D_T(i-1, j-1) + D_T(i-1, m-j)$$
$$+ D_T(n-i, j-1) + D_T(n-i, m-j) \tag{86}$$

and

$$S(i,j) = \sum_{i'=1}^{i-1} i' + \sum_{i'=1}^{n-i} i' + \sum_{j'=1}^{j-1} j' + \sum_{j'=1}^{m-j} j' \tag{87}$$

Thus, the sum of the distance between all pairs of the nodes is

$$D_T = \sum_{i=1}^{n} \sum_{j=1}^{m} D_T(i,j) \tag{88}$$

Using Equations (85) and (86), the solution to Equation (88) is

$$D_T = \frac{1}{3}(n^3m^2 + n^2m^3 - n^2m - nm^2) \tag{89}$$

The number of node-pairs in the lattice is $nm(nm - 1)$. Therefore, the characteristic-path length of the rectangular lattice, $L_N = \frac{1}{3}(n + m)$. Let $\Delta_{mn} = |n - m|$, then the

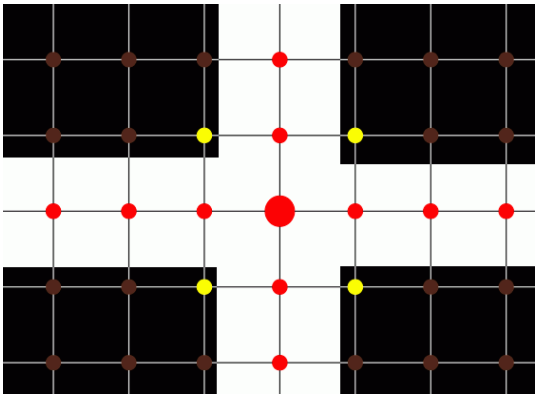

Figure 6.2. Pictorial depiction of Equation (86)**.**

characteristic-path length is

$$L_N = \frac{2}{3}\sqrt{|V| + \frac{1}{4}\Delta_{mn}^2} \tag{90}$$

The increase in L_N by order of $|V|^{1/2}$ for a two dimensional lattice conforms with the increase found in [56]. This value is also a loose upper bound for characteristic-path length of a triangular lattice. A loose lower bound can be obtained by changing the coefficient of $(ij - i - j + 1)$ in Equation (83) from 2 to 1. This suggests there is a direct link between nodes (i, j) and $(i - 1, j - 1)$.

6.1.2 Clustering coefficient

Since no triangles exist in a regular rectangular lattice, the average clustering of the network is 0. To maintain planarity, there can at most be one diagonal link in any of the rectangles of a triangular lattices. Hence, the number of neighbors of a node is defined by binomial probability distribution; i.e., inside a given rectangle a diagonal can take one of the two possible orientations. For very large networks all nodes can be assumed to be inside the lattice boundaries. Thus every node belongs to 4 rectangles and have neighbors ranging from 4 to 8. In each rectangle there is one diagonal. If the diagonal has its end

on the node of interest, the node belongs to two triangles and otherwise to one. Therefore based on the orientation of the four possible diagonals a node can belong to 4 to 8 number of triangles as shown in Figure 6.3. The number of triangles incident on the node follows a binomial probability distribution.

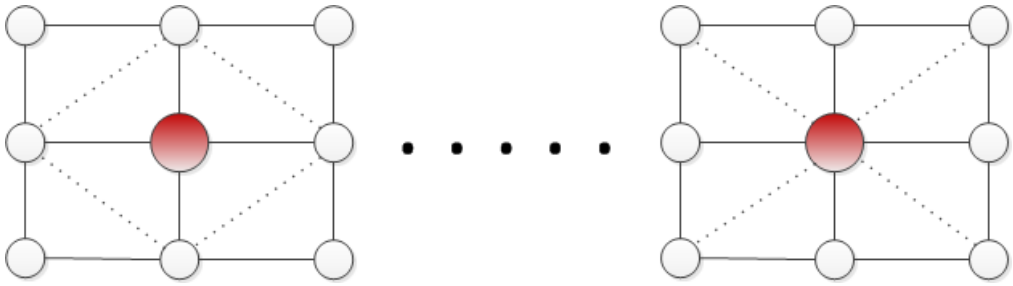

Figure 6.3. Clustering index: node has (a) four and (b) eight triangles incident on it.

Let $X_i \in \{0, 1, 2, 3, 4\}$ be the number of diagonal links incident on the node of interest, i, then $T_i \in \{4, 5, 6, 7, 8\}$ is the corresponding number of triangles incident to the node. Since orientations of the links inside all the four rectangles are independent of each other with each probability, $p = 1/2$, the probability of finding x diagonal links incident to i is

$$\Pr(X_i = x) = \binom{4}{x}(\frac{1}{2})^x(1-\frac{1}{2})^{4-x} = \frac{1}{16}\binom{4}{x} \tag{91}$$

For $X_i = x$, the degree and the number of triangles incident to i are both $4 + x$,, thus the clustering coefficient of i (the ratio of existing triangles incident on i to possible triangles that i can be a vertex to) is $C(x) = \frac{4+x}{\binom{4+x}{2}}$. Therefore, the average network clustering is

$$\begin{aligned} C = E[C(x)] &= \sum_{k=0}^{4} C(x)\Pr(X = x) \qquad (92) \\ &= \frac{1}{16}\sum_{x=0}^{4}\binom{4}{x}\frac{4+x}{\binom{4+x}{2}} \approx 0.42 \end{aligned}$$

For large networks, the number of non-boundary nodes is significantly higher than the

number of boundary nodes. Therefore the network's average clustering is $C \approx 0.42$.

6.2 Small-world properties of networks built on grid lattices

The lattice structures discussed so far are only connected to their immediate neighbors. To obtain small-world networks from these lattices, rewiring some of the links is necessary. Rewiring is the process of replacing one end node of a link by another node. The clustering coefficient of planar rectangular lattices is 0. Since small-world networks are expected to be highly clustered, obtaining a small-world network by rewiring a rectangular lattice is highly improbable. Hence it is better to study the small-world property of a triangular lattice.

Let link (u, v) connect nodes u and v, then with a given rewiring probability, μ, the link (u, v) is replaced by link (u, w) where w is an arbitrary node that is not connected to node u. Let $L_{N\mu}$ be the characteristic-path length and C_μ be the average clustering of the rewired network, and let L_{No} and C_o be the corresponding values in the original network. Then for a small ϵ, $\mu_l(\epsilon)$ is defined as the lowest rewiring probability such that $L_{N\mu}/L_{No}$ is less than ϵ. Similarly, $\mu_u(\epsilon)$ is the highest rewiring probability such that C_μ/C_o is greater than $1-\epsilon$. The value of ϵ and the range $[\mu_l(\epsilon), \mu_u(\epsilon)]$ indicate the emergence small-world property [56]. Figure 6.4 shows the impact of rewiring on a network with $|V| = 4000$.

Figure 6.4a shows a range of rewiring probability ≈ 0.05 for $\epsilon = 0.3$. These values are significantly different from those of 1-D lattices which show strong small-world property [51]. The example 1-D lattice rewiring shown in Figure 2.2 has a range of rewiring probability ≈ 0.05 for $\epsilon = 0.2$. A triangular 2-D lattice has six neighbors on average and 0.42 average clustering, and its characteristic-path length is less than $\sqrt{|V|}$. On the

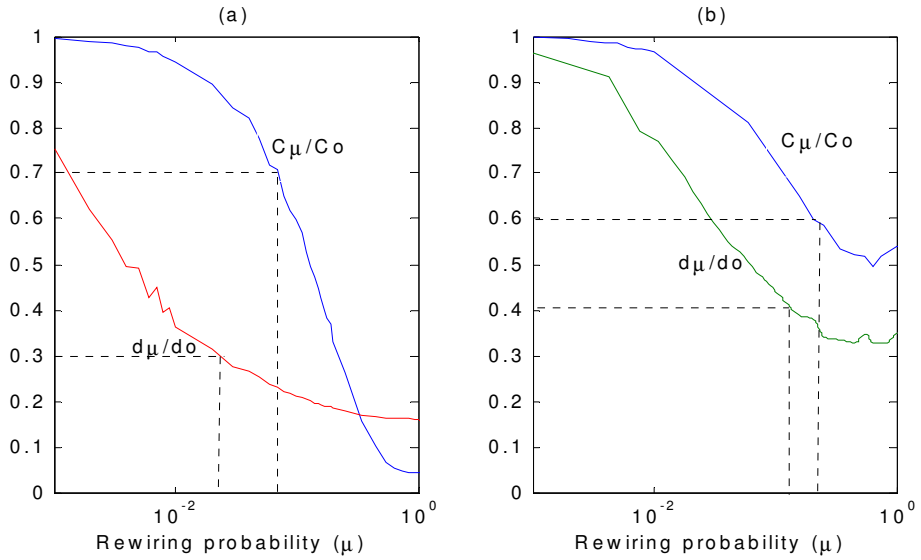

Figure 6.4. Characteristic path length and clustering of a network versus rewiring probability (a) random rewiring (b) distance based rewiring, $\alpha = 4$.

other hand, a 1-D lattice with average degree of six has an average clustering of 0.6 and a characteristic-path length of $|V|/12$ approximately. The characteristic-path length of random networks is function of $\ln|V|$. Therefore, $L_{N\mu}/L_{No}$ decays at a much lower rate when rewiring triangular lattices than 1-D lattices forcing ϵ to larger values and closing the $[\mu_l(\epsilon), \mu_u(\epsilon)]$ span.

A jump from one end of a network to another end via rewiring is possible in random reconnection. This reduces the characteristic-path length significantly. Although this might be the case in social networks, building such type of links in actual infrastructure networks is less practical. Therefore, jumps are limited to nodes that are a short lattice-distance away. Therefore, when replacing link (u, v) by (u, w), node w is selected based on a probability distribution built on the lattice distance. A lattice distance between node u and v is the number of hops that takes to reach v from u in the original full lattice.

In this work, a power-law function is used, similar to Kleinberg's navigability model [90], to favor near nodes while ensuring all nodes have a chance to be at the end of the rewired link. Let d_{uw} be the lattice distance between nodes u and w, then the probability of the new link (u, w) having d_{uw} lattice distance is given by $p(d_{uw}) = Ad_{uw}^{-\alpha}$, where A is normalization constant and α is the decay factor.

In distance-dependent rewiring, since the lattice distance of the shortcuts is limited, the expected rate of decrease of characteristic-path length and clustering is smaller than the randomly rewired case as shown in Figure 6.4b.

6.3 Small-world highway network

To study the small-world property in an actual road network, rewiring is performed on part of the North Carolina road network. The network is obtained from TIGER/Line road network database, as provided by the Ninth DIMACS Implementation Challenge[2]. Specifically, the area around the Research Triangle Park (RTP) is selected. The selected network consists of 20,000 nodes and has an $|E|/|V|$ ratio of 2.38. The road network is shown in Figure 6.5. A closer view shows that there are much more rectangles than triangles, hence low clustering. The road network of Research Triangle Park, NC.

The RTP network has a characteristic-path length $L_N = 44$ hops and an average clustering $C = 0.019$ respectively. The random network equivalent of this network has $L_N = 11.12$ and $C = 3 * 10^{-4}$. The lattice equivalent of the network has $L_N = 105.7$ and $C = 0.124$. Therefore the clustering and characteristic-path length ratios are $C_\mu/C_o = 0.15$ and $L_{N\mu}/L_{No} = 0.42$ respectively. From Figure 6.4 it is clear that there is no value of μ

[2] See http://dimacs.rutgers.edu/Workshops/Challenge9/

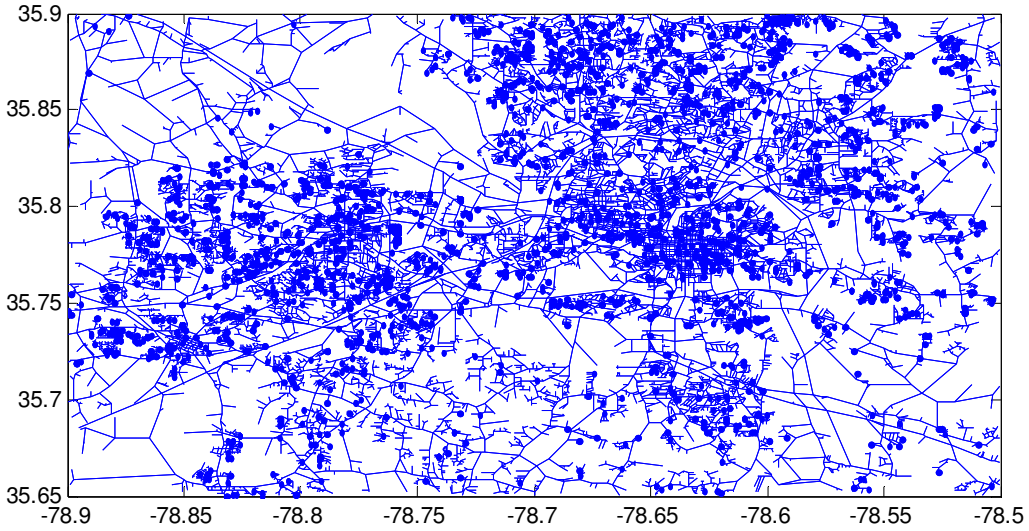

Figure 6.5. The road network of Research Triangle Park, NC.

in both (a) and (b) plots that satisfies both ratios. Hence, the small-world property is not observed at this granularity.

The network can be coarsened by using the coarsening procedures discussed in chapter 5. In this work, a 1:20 coarsening is used in this work. The coarsened network has 1000 nodes and $|E|/|V|$ ratio of 3.47.

The coarsened network has a 14 hops characteristic path length and 0.21 average clustering. An equivalent 2-D planar lattice of the coarsened network has 20 and 0.23 characteristic path length and average clustering paths respectively. A complete distance based rewiring of this lattice with $a = 4$ results in a network with $L_N = 5.5$ and $C = 0.002$. Therefore $C_\mu/C_o = 0.91$ and $L_{N\mu}/L_{No} = 0.7$. Figure 6.4b can be interpolated to return a μ value of 0.02 that satisfies both conditions. This indicates that a road network exhibits a small-world property.

CHAPTER 7

CONCLUSIONS

This work has addressed three issues of critical importance in analyzing the survivability of a network.

- Measures were developed to capture structural and functional survivability.
- Measures were proposed to quantify the propagation of impact of a fault across the network.
- A model was developed to reduce the complexity of large and complex networks.

Among several types of connectivity measures investigated, a measure that captures the essence of the survivability was developed as a connectivity-based robustness measure. A linear program that finds the shortest set of edge-disjoint paths was formulated and used to measure the connectivity of the network. The algorithm developed has a complexity of a single run of an integer linear program with $|V| + 1$ constraints and $|E| + k$ decision variables for a graph $G(V, E)$.

Localized large-scale faults exhibit different pattern than random single or even multiple nodal failures. Although multiple-failure scenarios can be combinatorial in nature, the geographical locality is shown to reduce the number of fault scenarios significantly. A model that captures the pattern of geographically localized large-scale fault in spatial networks is developed. A group centrality measure is developed to predict the response of a network to large-scale localized perturbation.

In Chapter 4, survivability measures that capture the propagation of impact of node failures throughout the network were proposed. This was achieved by introducing the

notion of local survivability and node-to-path distance.

Using local survivability measures, the following were observed experimentally;

- The expected decrease in loss of performance between node pairs as a result of single node failure is a function of the distance between the fault node and the node pair and the shortest path length of the node pair.
- The expected loss in max-flow capacity between two nodes decreases exponentially as a function of the distance between a fault node and the node pair.
- The expected flow and connectivity losses decay as a power of shortest path length.
- Node pairs with smaller shortest path length and closer to the fault node are more affected by a node failure.

To mitigate the computational burden of survivability analysis, a geographic-based coarse-graining procedure for network complexity reduction is proposed. The coarse graining strategy is implemented by building a graph-based representations of the interactions of a network partitions. The partitions are designed to be contiguous and geographically compact so that each partition behaves as a subnetwork. A significant complexity reduction in graph computations is achieved by partitioning the network into regions and applying a probabilistic model on the original and coarsened network. The probabilistic estimation model produced more accurate results for graph clustering than for characteristic-path length. The coarse graining strategy is also used to develop a hierarchical routing algorithm. The hierarchical algorithm is shown to find shortest paths that are comparable to the traditional shortest path algorithms.

The level of coarsening is found to affect the network measures. A series of valida-

tion tests were conducted using highway networks to confirm the validity of this coarsening approach:

- The characteristic-path length of a network decreased exponentially with an increase in the coarsening level while both the average clustering and the network connectivity increased.
- On average, the hierarchical route is only 23% longer than the shortest route.
- The worst estimation error of network measures found is 7.15%.

Another byproduct of the coarse graining strategy was the demonstration of small-world property in road networks. The results obtained from rewiring triangular planar lattices indicate emergence of small-world networks. Nevertheless, the small-world property observed is not as strong as 1-D ring lattices. This work demonstrated that a high-level (coarsened) view of the road network has a similar small-world property as a network that lies somewhere between a regular triangular 2-D lattice and a random geographically-localized network. It is also shown analytically that 2-D planar lattices have a characteristic-path length of order $|V|^{1/2}$ and maximum average clustering ≈ 0.42.

The experiments in this work were conducted using the highway network, but the conclusions on survivability and coarsening can be easily extended to include other spatial networks. This work can be improved by investigating optimal structural addition to increase local and global survivability. In addition, the hierarchical estimation and routing approaches developed using coarsening can be used to implement a hierarchical survivability enhancement scheme and to improve the performance of distributed adaptive routing algorithms.

BIBLIOGRAPHY

[1] R. Ellison, D. Fisher, R. Linger, H. Lipson, T. Longstaff, and N. Mead, Survivable Network Systems: An Emerging Discipline, Carnegie Mellon/Software Engineering Institute (Technical Report No. CMU/SEI-97-TR-013), 1997.

[2] S. Liew and K. Lu, A framework for characterizing disaster-based network survivability, *IEEE Journal on Selected Areas in Communications*, vol. 12, no. 1, 1994, pp. 52-58.

[3] H. Frank, Survivability Analysis of Command and Control Communications Networks–Part I, *IEEE Transactions on Communications*, vol. 22, no. 5, 1974, pp. 589- 595.

[4] P. Heegaard and K. Trivedi, Network survivability modeling, *Computer Networks*, vol. 53, 2009, pp. 1215-1234.

[5] D. Gfeller and P. De Los Rios, Spectral coarse-graining of complex networks. *Phys. Rev. Letters,* vol. 99, 2007, pp. 038701-4.

[6] B. Kim, Geographical Coarse Graining of Complex Networks. *Phys. Rev. Letters,* vol. 93, 2004, pp. 168701-4.

[7] R. Tarjan, Depth first search and linear graph algorithms, *SIAM Journal on Computing*, vol. 1, 1972, pp. 146-160.

[8] E. Dijkstra, A Note on Two Problems in Connection with Graphs, *Numerishce Mathematic*, 1959.

[9] R. Bhandari, *Survivable Networks: Algorithms for Diverse Routing*, Kluwer Academic Publishers, 1999.

[10] T. Cormen, C. Leiserson, and R. Rivest, *Introduction to Algorithms*, The MIT Press, 2000.

[11] S. Pallottino and M. Scutella, Shortest Path Algorithms in Transportation Models: Classical and Innovative Aspects, Universita di Pisa, (Technical Report: TR-97-06), 1997.

[12] M. Stoer, *Design of survivable networks*, Lecture Notes in Mathematics, vol. 1531, Springer-Verlag, 1992.

[13] Y. Kobayashi and C. Sommer, On shortest disjoint paths in planar graphs, *Discrete Optimization*, vol. 7, 2010, pp. 234-245.

[14] J. Suurballe, Disjoint Paths in a Network, *Networks*, vol. 4, 1974, pp. 125-145.

[15] L. Ford and D. Fulkerson, Maximal Flow Through a Network, *Canadian Journal of Mathematics*, vol. 8, 1956, pp. 399-404.

[16] R. Potss, R. Oliver, *Flows in Transportation Network*, Academic Press, 1972.

[17] Wai-Kan Chen, *Theory of Nets: Flows in Networks*, John Wiley & Sons, Inc. , 1990.

[18] D. B. West, *Introduction to Graph Theory*, Prentice-Hall, Inc, 1996.

[19] D. Koschutzki, K. Lehmann, L. Peeters, S. Richter, D. Renfelde-Podehl, and O. Zlotowski, Centrality Indices, in *Network Analysis: Methodological Foundations*, Springer-Verlag, 2005, pp. 16-61.

[20] S. Borgatti and M. Everett, A Graph-Theoretic Perspective on Centrality, *Social Networks*, vol. 28, 2006, pp. 466-484.

[21] P. Hage, F. Harrary, Eccentricity and centrality in networks, *Social Networks*, vol. 17, 1995, pp. 57-63.

[22] P. Bonacich, Some unique properties of eigenvector centrality, *Social Networks*, vol. 29, 2007, pp. 555-564.

[23] S. Borgatti, Centrality and network flow, *Social Networks*, vol. 27, 2005, pp. 55-71.

[24] S. Wuchty and P. Stadler, Centers of complex networks, *Journal of Theoretical Biology*, vol. 223, 2003, pp. 45-53.

[25] E. Estrada and N. Hatano, Statisitcal-mechanical approach to subgraph centrality in complex networks, *Chemical Physics Letters*, vol. 439, 2007, pp. 247-251.

[26] Z. Guan, L. Chen, and T. Qian, Routing in scale-free networks based on expanding betweenness centrality, *Physica A*, vol. 390, 2010, pp. 1131-1138.

[27] I. Mishkovski, M. Biey, and L. Kocarev, Vulnerability of complex networks, *Communications in Nonlinear Science and Numerical Simulation*, vol. 16, 2011, pp. 341-349.

[28] E. Zio and R. Piccinelli, Randomized flow model and centrality measure for electrical power transmission network anlaysis, *Reliability engineering and System Safety*, vol. 95, 2010, pp. 379-385.

[29] P. Crucitti, V. Latora, and S. Porta, Centrality Measures in Spatial Networks of Urban Streets, *Phys. Rev. E*, vol. 73, 2006, pp. 036125-9.

[30] E. Strano, A. Cardillo, V. Lacoviello, et al., Street Centrality vs. Commerce and Service Locations in Cities: a Kernel Density Correlation case in Bologna, Italy, ArXiv Physics e-prints, 2007

[31] U. Demsar, O. Spatenkova, and K. Virrantaus, Identifying Critical Locations in a Spatial Network with Graph Theory, *Transactions in GIS*, vol. 12, 2008, pp. 61-82.

[32] M. Fiedler, A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory, *Czechoslovak Mathemtaical Journal*, vol. 25, 1975, pp. 619-633.

[33] A. Berman, R. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, SIAM, 1994

[34] U. Brandes, On variants of shortest-path betweenness centrality and their generic computation, *Social Networks*, vol. 30. 2008, pp. 136-145.

[35] E. Estrada, D. Hingham, and N. Hatano, Communicability betwneenness in complex networks, *Physica A*, vol. 388,2009, pp. 764-774.

[36] V. Latora, M. Marchiori, A measure of Centrality based on network efficiency, *New Journal of Physics*, vol. 9, 2007 pp. 188.

[37] G. Fagiolo, Clustering in complex directed networks, *Phys. Rev. E.*, vol. 76, 2007, pp. 026107.

[38] D. Barmpoutis and R. Murray, Networks with the Smallest Average Distance and the Largest Average Clustering, arXiv:1007. 4031v1, 2010.

[39] J. Kruskal, On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem, *Proceedings of the American Mathematical Society*, Vol. 7, 1956.

[40] R. Prim, Shortest Connection Networks and some Generalizations, *Bell Systems Technical*

Journal, 1957.

[41] J. Ullman, An Algorithm for Subgraph Isomorphism, *Journal of the Association for Computing Machinery*, 1976.

[42] P. Foggia, C. Sansone, and M. Vento, A Performance Comparison of Five Algorithms for Graph Isomorphism, *Proceedings of the 3rd IAPR TC-15 Workshop*, 2001, pp. 188-199.

[43] A. Baltz and L. Kliemann, Spectral Analysis, in *Network Analysis: Methodological Foundations*, Springer-Verlag Berlin Heidelberg, 2005

[44] M. Fiedler, Algebraic connectivity of graphs, *Czechoslovak Mathematical Journal*, 1973

[45] S. Guattery and G. Miler, On the Performance of Spectral Graph Partitioning Methods, Proceedings of ACM-SIAM Symposium, 1995, pp. 233-242.

[46] P. Chan, M. Schlag, and J. Zien, Spectral K-Way Ratio-Cut Partitioning and Clustering, *IEEE transaction on computer-aided design of integrated circuits and systems*, vol. 13, 1994, pp. 1088-1096.

[47] P. Erdos and A. Reny, On Random Graphs, Publicationes mathematicae, vol. 6, 1959, pp. 290-297.

[48] E. Gilbert, Random graphs, *Annals of Mathematical Statistics*, vol. 30, 1959, pp. 1141-1144.

[49] S. Milgram, The small-world problem, *Psychology Today*, vol. 1, 1969, pp. 61-67.

[50] J. Travers and S. Milgram, An Experimental study of small world problem, *Sociometry*, vol. 32,1969, pp. 425-443.

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

[52] L. Amaral, A. Scala, M. Barthelemy, and H. Stanley, Classes of small-world networks, *PNAS*, vol. 97, 2000, pp. 11149-11152.

[53] L. A. Adamic, The small world web, in Lecture Notes in Computer Science, vol. 1696, 1999, Springer, pp. 443-454.

[54] V. Latora and M. Marchiori, The Architecture of Complex Systems, in *Nonextensive Entropy Interdisciplinary Applications*, Oxford University Press, 2004.

[55] D. J. Watts, *Small worlds: the dynamics of networks between order and randomness*, Princeton University Press, 1999

[56] F. Vega-Redondo, *Complex Social Networks*, Cambridge University Press, 2007

[57] P. Blanchard and T. Krueger, Small world graphs by iterated local edge formation, *Phys. Rev. E*, vol. 71, 2005, pp. 046139.

[58] M. E. J. Newmann, I. Jensen, and R. M. Ziff, Percolation and epidemics in a two-dimensional small world, *Phys. Rev. E*, vol. 65, 2002, pp. 021904.

[59] S. Morris, Contagion, *Review of Economic Studies*, vol. 67, 2000, pp. 57-78.

[60] A. Barabasi and R. Albert, R, Emergence of Scaling in Random Networks, *Science*, vol. 286, 1999, pp. 509-512.

[61] B. Hendrickson and R. Leland, A multilevel algorithm for partitioning graphs, Technical

Report SAND93-1301, Sandia National Laboratories, 1993

[62] M. Girvan and M. Newman, Community structure in social and biological networks, *PNAS,* vol. 12, 2002, pp. 7821-7826.

[63] R. Geisberger, P. Sanders, D. Schultes, and D. Delling, Contraction Hierarchies: Faster and Simpler Hierarchical Routing in Road Networks, *Proc. Experimental Algorithms (WEA'08)*, LNCS, vol. 5038, 2008, pp. 319-333.

[64] G. Marshall and K. Erciyes, Implementation of a Cluster Based Routing Protocol for Mobile Networks, *ICCS*, LNCS, vol. 3514, 2005, pp. 388-395.

[65] J. Flich, A. Mejia, P. Lopez, and J. Duato, Region-based routing: an efficient routing mechanism to tackle unreliable hardware in network on chips, *Proc. IEEE NOCS*, 2007, pp. 183-194.

[66] B. Kernighan and S. Lin, An Efficient Heuristic Procedure for Partitioning Graphs, *Bell System Technical Journal,* vol. 49, 1970, pp. 291-307.

[67] J. Pinney and D. Westhead, Betweenness-Based Decomposition Methods for Social and Biological Networks., in *Interdisciplinary Statistics and Bioinformatics,* Leeds University Press, 2006.

[68] S. Hayasaka and P. Laurienti, Comparison of characteristics between region-and voxel-based network analyses in resting-state fMRI data., *NeuroImage*, vol. 50, 2010, pp. 499-508.

[69] M. Shelley and D. McLaughlin, Coarse-Grained Reduction and Analysis of a Network Model of Cortical Response: I. Drifting Grating Stimuli, *Journal of Computational Neuroscience*, vol. 12, 2002, pp. 97-122.

[70] H. Conzelman, D. Frey, and E. Gilles, Exact model reduction of combinatorial reaction networks, *BMC Systems Biology*, vol. 2, 2008, pp. 78-102.

[71] C. Chennubholta and I. Bahar, Markov methods for hierarchical coarse-graining of large protein dynamics, *Journal of computational biology*, vol. 14, 2007, pp. 765-776.

[72] H. Kang and T. Kurtz, Separation of time-scales and model reduction for stochastic reaction networks, arXiv:1011. 1672v1 [math. PR].

[73] G. Karypis and V. Kumar, Multilevel K-Way Partitioning Scheme for Irregular Graphs, *Journal of Parallel and Distributed Computing*, vol. 48, 1998, pp. 96-129.

[74] S. Lafon and A. Lee, Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization., *IEEE transaction on pattern analysis and machine intelligence,* vol. 28, 2006, pp. 1393-1403.

[75] D. Spielman and S. Teng, A LocaL Clustering Algorithm for Massive Graphs and its Application to Nearly-Linear Time Graph Partitioning. arXiv:0809. 3232, 2008.

[76] H. Yu, Z. Hongli, Survivability performance evaluation for satellite communication network based on walker constellation, Proc. of SPIE, vol. 6795, 2007.

[77] S. D. Nikolopoulos, A. Pitsillides, and D. Tipper, Addressing Network Survivability Issues by Finding the K-best Paths through a Trellis Graph, *Proceedings of IEEE computer and communications societies*, vol. 1, 1997, pp. 370-377.

[78] H. Kang, C. Butler, Q. Yang, and J. Chen, A new survivability measure for military communication networks, *IEEE-MILCOM*, vol. 1, 1998, pp. 71-75.

[79] K. Kim, B. Roh, Y. Ko, W. Choi, and E. S. Son, Survivability measure for multichannel MANET-based tactical networks, *Proc. of International Conference on Advanced Communication Technology*, 2011, pp. 1049-1053.

[80] H. Frank, Survivability Analysis of Command and Control Communications Networks–Part II, *IEEE Transactions on Communications*, vol. 22, 1974, pp. 596- 605.

[81] W. Pullan, Optimizing Multiple Aspects of Network Survivability, *Proc. of Evolutionary Computation*, vol. 1, 2002, pp. 115-120.

[82] W. Molisz, Survivability Function - A Measure of Disaster-based routing performance, *IEEE journal on selected areas in communications*, vol. 22, 2004, pp. 1876-1883.

[83] Y. Liu and K. S. Trivedi, A general framework for network survivability quantification, *Proc. 12th GI/ITG Conf. measuring, Modeling and Evaluation of Computer and Communication Systems*, 2004.

[84] L. Zhang, W. Wang, L. Guo, W. Yang, and Y. Yang, A survivability Quantitative Analysis model for network system based on attack graph, *Proceedings of Machine Learning and Cybernetics*, vol. 6, 2007, pp. 3211-3216.

[85] C. Fung, Y. Chen, X. Wang, J. Lee, R. Tarquini, M. Anderson, Survivability analysis of distributed systems using attack tree methodology, MILCOM, IEEE, 2005

[86] L. Beineke, O. Oellermann , and R. Pippert, The average connectivity of a graph, Discrete Math. 252 (2002) 31-45.

[87] R. Ahuja, T. Magnanti, and J. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, 1993

[88] A. Sen, B. Shen, L. Zhou, and B. Hao, Fault-Tolerance in Sensor networks: A New Evaluation Metric, *Proc. of INFOCOM 2006,* 2006, PP. 1-12.

[89] G. Karypis, V. Kumar, METIS* A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices Version 4. 0, University of Minnesota, Department of Computer Science / Army HPC Research Center, 1998

[90] J. Kleinberg, Navigation in a small world, *Nature*, vol. 406, 2000, pp. 845.