On the Evolution of
Finite Sized Complex Networks

A thesis submitted
in partial fulfillment for the degree of

Doctor of Philosophy

by

ABHISHEK CHAKRABORTY

DEPARTMENT OF AVIONICS

INDIAN INSTITUTE OF SPACE SCIENCE AND TECHNOLOGY

THIRUVANANTHAPURAM - 695547, INDIA

MAY 2018

Certificate

This is to certify that the thesis entitled **"On the evolution of finite sized complex networks"** submitted by **Abhishek Chakraborty**, to the Indian Institute of Space Science and Technology, Thiruvananthapuram, in partial fulfillment for the award of the degree of **Doctor of Philosophy**, is a *bona fide* record of the research work carried out by him under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institution or University for the award of any degree or diploma.

Dr. B. S. Manoj
Supervisor
Professor and Head
Department of Avionics

Counter signature of HOD with seal

Place: IIST, Thiruvananthapuram
Date: May 2018

Declaration

I declare that this thesis titled **"On the evolution of finite sized complex networks"** submitted in partial fulfillment for the award of the degree of **Doctor of Philosophy** is a record of original research work carried out by me under the supervision of **Dr. B. S. Manoj**, and has not formed the basis for the award of any degree, diploma, associateship, fellowship, or other titles in this or any other Institution or University of higher learning. In keeping with the ethical practice in reporting scientific information, due acknowledgments have been made wherever the findings of others have been cited.

Abhishek Chakraborty
Research Scholar
Department of Avionics
Roll No.: SC12D011

Place: IIST, Thiruvananthapuram
Date: May 2018

Acknowledgments

This PhD thesis work owes its existence to many persons without whose help it would not have been possible to complete.

First and foremost, I am very grateful to my dear supervisor Dr. B. S. Manoj for his encouraging attitude, kind-hearted support, indispensable comments, and critical insights throughout the thesis work. He has been a faculty with good grasp of the subject, an excellent researcher, and a sought after philosopher. I will admire him always.

Beside my PhD thesis supervisor, I would like to thank my doctoral committee members: Dr. N. Selvaganesan (Committee chair), Dr. Priyadarshan Hari, Dr. Anil Kumar C. V., Dr. Bheemarjuna Reddy Tamma (IIT Hyderabad), Dr. Venkata Ramana Badarla (formerly with IIT Jodhpur and presently with IIT Tirupati), and Dr. Shine Lal E. (Government College Chittur, Palakkad), for their constant inspiration, thought-provoking comments, and critical technical reviews throughout the thesis work.

I would also like to thank Dr. Vineeth B. S. for his kind and enthusiastic support, valuable advice, and esteemed guidance during the period of this thesis work. Many long hours of technical discussion with Dr. Vineeth have immensely helped me developing insights to address several challenging problems.

Words are not enough to thank my lab-mate Sarath Babu for our countless discussions on technical and many non-technical matters over the years at the Systems and Networks Lab. I am immensely grateful to Sarath for educating me on *how to think like a computer scientist*.

Moreover, I would also like to thank Rahul Singh, a former MTech student and Dr. Dharmendra Singh Yadav, former post-doctoral fellow for numerous interesting discourses during their stay in our lab. Furthermore, my wholehearted thanks go to all internship and BTech final year project students in our lab and not only limited to Nivedita Gaur, Arun K. P., Priti Singh, Ravi Teja Naidu, Abhishek Bhaumick, Pramod Reddy, Y. Naga Rahul, Keshav Dayal, Kolli Aravind, Gautham Suresh, and V. Mohanasruthi.

I would like to thank Dr. Deepak T. G. and Dr. Thomas Kurian for their kind cooperation and valuable advices. I would also like to deliver my kind gratitude and respect to Dr. Raju K. George, Dean R&D, Dr. Kuruvilla Joseph, Dean Student Activities, Dr. Kurien Issac, Dean Intellectual Property Rights & Continuing Education, and Dr. A. Chandrasekar, Dean Academic and the Registrar for providing me adequate facilities that enabled me to complete my PhD thesis. Furthermore, I am immensely thankful to Dr. V. K. Dadhwal, the Director and Dr. K. S. Dasgupta, the former Director for providing me a wonderful opportunity for pursuing PhD at IIST.

My sincere thanks also go to all the staff members of Avionics (technical and non-technical), especially Ms. Divya R. S. and Ms. Preetha T., as well as IIST library and information services for their direct or indirect kind support during the course of the PhD thesis. Further, I would like to thank IIST administration, hostel services, transport services, cleaning and maintenance division, sports facility, and canteen services for assuring my pleasant and memorable stay at IIST.

The long stay-away from home was very challenging and sometimes seemed to be impossible. At that time a support system was needed to overcome that situation. I am very fortunate that I have got four saviors, namely Najeeb P. K., Rakesh R., Sujith Vijayan, and Swagat Ranjan Das, throughout the entire period of stay at IIST. Further, it will be unfair if I do not thank Ameya Anil Kesarkar, Raja J., Rahul G. Waghmare, Arun Prasad K., Mathiazhagan S., Sabu M., Rajesh R., Deepak M., Dibyendu Adak, P. Suresh Kumar, Deepak G., Richu Sebastian, Rajkumar R., Praveen Wilson, Nikhilraj A., Mahesh T. V., and Praveenkumar K., for their unconditional companionship.

Last but certainly not the least, I would like to thank my family members who not only taught me how to rise, but also became the wind beneath my wings. I am very grateful to my parents Smt. Sima Chakraborty and the late Shri. Bikash Chakraborty, for their fantastic upbringing and educating moral values to help transforming me to a nice human being. I would not have made this so far without their love, guidance, support, and most importantly, their wholehearted prayers.

Abhishek Chakraborty

IIST, Thiruvananthapuram.

Abstract

Any complex physical system, man-made or natural, consists of entities each of which interacts with other entities in the system. Such complex systems can be modeled as network graphs where the entities are nodes and their interactions are edges of the network graph. Earlier studies reported possible mechanisms for the evolution of complex networks where size of the network is growing, in the context of nodes and edges, with time. To the best of our knowledge, the characteristics of finite sized complex systems, which can be seen in many real-world networks, such as relationships in community networks, transportation networks, and wireless sensor networks, are not studied in depth. Here, the finite sized networks mean that such complex physical systems are not growing in size when the total number of nodes is concerned. This thesis aims to study the reasoning behind the evolution of such finite sized complex networks.

We find that the greedy decision making, based on the optimization of certain network metrics, results in unique structural characteristics during the evolution of many complex networks. In a finite sized complex network, minimization of the end-to-end hop distance using the optimal/near-optimal long-ranged link (LL) addition for minimizing the average path length (APL), maximizing the centrality measures, or maximizing the overall network flow capacity, constitutes the greedy decision making. It is also observed that when LLs are added optimally/near-optimally, e.g., by minimizing the APL, the resulting network evolves to a scale-free network with a few hub nodes where a large number of LLs are incident.

To study the greedy optimal/near-optimal decision based network evolution, we consider addition of new LLs in a finite sized string topology network with the greedy near-optimal decision to minimize the APL of the string network which can be considered to be one of the most sparse regular network model. We observe that, in an N-node string topology network, the first LL is always optimally connected between the anchor nodes

at the $0.2N^{th}$ and $0.8N^{th}$ fractional locations. The fixed fractional locations of the anchor nodes also have been analytically found at 0.2071 and 0.7929.

We then consider a model motivated by practical limitations, where constraints are placed on the length of the LLs. It is found, in a finite sized complex network, with the optimal addition of length constraint LLs, that there is a visible transition of a fixed sized regular network in the following manner: from a regular network to a small-world network, then to a scale-free network with the truncated degree distribution, and at last, to a fully connected network.

As the greedy decision based LL addition is computationally intensive, a heuristic approach, sequential deterministic LL addition (SDLA) is also proposed in the context of unweighted string network, to efficiently transform the network to a small-world network. SDLA algorithm can help efficient design and deployment of moderate sized string topology networks for various applications, such as community broadband networks, computer networks, tactical networks, and emergency response networks.

Next, we apply our above observations to transform a finite sized string topology wireless sensor network to an APL to the base station (BS) optimal (APLB-optimal) small-world wireless sensor network by introducing a few LLs. The optimal LL addition also incorporates tradeoffs between the excess transmission power and the overall path length reduction. Our analytical observations on the locations of newly added links (single and two LLs) also satisfy the simulation and the approximate observations.

To the end of this thesis, we propose an exhaustive search based LL addition algorithm, maximum flow capacity (MaxCap) that deterministically maximizes the average network flow capacity (ANFC) in a weighted undirected network. Based on the observations from MaxCap, we propose a new link addition heuristic, average flow capacity enhancement using small-world characteristics (ACES), that improves the end-to-end distance traversed by incorporating the small-world characteristics, and also enhances the overall performance of a network. We also validate our observations through exhaustive simulations on various real-world road networks. ACES can find many real-world applications in communication networks, transportation networks, and tactical networks where ANFC is a very critical parameter.

Contents

List of Tables xv

List of Figures xvii

List of Algorithms xxi

Acronyms xxiii

Notation xxv

1 Introduction 1

1.1 Motivation and Objectives 1

1.2 Major Contributions of the Thesis 3

1.3 Thesis Outline 5

1.4 Summary 6

2 Introduction to Complex Networks 9

2.1 Real-World Complex Networks 9

2.1.1 The Author Citation Networks 10

2.1.2 The Autonomous Systems in the Internet 10

2.1.3 The Air Traffic Networks 11

2.2 Complex Network Metrics 12

2.2.1 Average Nodal Degree 13

2.2.2 Average Clustering Coefficient 13

2.2.3 Average Path Length 15

2.2.4 Network Diameter 15
2.2.5 Degree Distribution 16
2.2.6 Centrality Metrics 16
2.3 Complex Network Models 21
2.3.1 Regular Networks 22
2.3.2 Random Networks 22
2.3.3 Small-World Networks 23
2.3.4 Scale-Free Networks 28
2.4 Summary 31

3 Greedy Link Addition in Finite Sized Networks **33**
3.1 Existing Literature 33
3.1.1 Creation of Random Small-World Networks 34
3.1.2 Creation of Deterministic Small-World Networks 35
3.2 Random LL Addition 37
3.2.1 Nodal Degree Distribution with Random LL Addition 38
3.3 Greedy Decision based LL Addition 41
3.3.1 Greedy Optimal LL Addition 41
3.3.2 Greedy Near-Optimal LL Addition 43
3.3.3 Degree Distribution 45
3.4 Length Constrained LL Addition 48
3.4.1 Constrained LL Addition in Finite String Topology Networks . . 49
3.4.2 Constrained LL Addition in Finite Grid Topology Networks . . 52
3.5 Summary 55

4 Analytical Identification of Anchor Nodes **57**
4.1 Existing Literature 57

4.2 Significance of Anchor Nodes 59
4.3 Identification of Anchor Nodes 59
4.3.1 Problem Statement 60
4.3.2 Anchor Nodes for a Dense String Network 61
4.4 On the Locations of Anchor Nodes 65
4.5 Influence of the Anchor Nodes 67
4.5.1 Based on Random LL Addition 67
4.5.2 Based on Greedy Near-Optimal LL Addition 67
4.5.3 Based on Combination of Both LL Addition 69
4.6 Summary 70

5 Sequential Deterministic Long-Ranged Link Addition **71**
5.1 Observations from Previous Study 71
5.2 LL Addition with SDLA Heuristic 72
5.2.1 The SDLA Algorithm 73
5.3 Performance Analysis of SDLA Algorithm 76
5.3.1 Average Path Length Reduction with SDLA 76
5.3.2 Average Clustering Coefficient and Centrality Measures 78
5.4 Observations and Discussion 80
5.5 Summary 80

6 Optimal Link Addition in Wireless Sensor Networks **81**
6.1 Existing Literature 81
6.2 System Model and Link Addition Problem 83
6.3 A Dense Graph Approximation: Analytical Solution 84
6.3.1 Case where $\mathbf{M=1}$ 85
6.3.2 Case where $\mathbf{M=2}$ 89

6.4 Summary 92

7 Achieving Capacity-Enhanced Small-World Networks 93

7.1 Existing Literature 93

7.2 Average Network Flow Capacity 95

7.2.1 Max-Flow Min-Cut Theorem 95

7.3 Link Addition with Maximum Flow Capacity 99

7.3.1 Time Complexity of MaxCap Algorithm 100

7.4 Average Flow Capacity Enhancement using Small-World Characteristics 101

7.4.1 Time Complexity of ACES 102

7.4.2 Why does ACES Work? 103

7.5 Performance Evaluation of ACES 103

7.5.1 Arbitrary Networks 104

7.5.2 Real-world Road Networks 107

7.6 Observations and Discussion 116

7.7 Summary 117

8 Conclusions and Future Research Directions 119

8.1 Conclusions 119

8.2 Future Research Directions 120

Bibliography 123

Publications based on the Thesis 131

Other Related Publications 133

Doctoral Committee and External Reviewers for PhD Research Review 135

List of Tables

2.1 Degree centrality 18

2.2 Closeness centrality 19

2.3 Betweenness centrality 20

2.4 Comparison of the three types of networks 24

2.5 From a 4-regular network to a small-world network with rewiring 26

3.1 Betweenness centrality values after the addition of each LL with the pure random LL addition 40

3.2 Betweenness centrality values after the addition of each LL with the greedy near-optimal LL addition 47

4.1 Total path length experienced by different parts of STNs 66

4.2 APL values for different positions of anchor nodes in STNs 69

5.1 Reduction of APL values with various numbers of LLs 77

5.2 Observations on graph centrality measures for various sized STNs . . . 79

6.1 Simulation and analytical observations on location of a single optimal LL 88

6.2 Simulation and analytical observations on locations of two optimal LLs 91

7.1 All possible flow capacities between a source node and a destination node 97

7.2 Characteristic features of six real-world road networks 108

7.3 Various road-types and corresponding flow capacity values 108

7.4 ACES performance with r-hop neighbors: LL with fixed flow capacity . 114

7.5 ACES performance with r-hop neighbors: LL with variable flow capacity 115

List of Figures

2.1 The author citation network of scientific work on network theory. 11

2.2 The network of autonomous systems in the Internet. 12

2.3 The air traffic network. 13

2.4 Example graphs for ACC calculation. 14

2.5 Examples of normal and power-law distributions. 16

2.6 An example network and its adjacency matrix. 17

2.7 A sample network and its cost matrix. 19

2.8 An example network to calculate betweenness centrality. 20

2.9 An example 4-regular network of 50 nodes. 22

2.10 An arbitrary random network model. 23

2.11 An example realization of the small-world characteristics with rewiring. 25

2.12 Applying rewiring strategy in regular grid network. 27

2.13 The Euclidean distance based link addition strategy in a 2D lattice. . . . 28

2.14 An example scale-free network of 50 nodes. 29

3.1 Pure random LL addition in a string topology network. 38

3.2 Degree distribution after LL addition with the pure random decision. . . 39

3.3 Greedy near-optimal LL addition to add six LLs in a 40-node STN. . . . 44

3.4 Degree distribution after greedy near-optimal LL addition strategy. . . . 46

3.5 Plot of D_{max}, D_{max2}, and D_{max3} with respect to different LL_{MaxLen} values in various sized STNs. 50

3.6 Plot of D_{avg} with respect to different LL_{MaxLen} values in various sized STNs. 51

3.7 Plot of D_{max}, D_{max2}, and D_{max3} with respect to different $LL_{MaxLen2D}$ values in various sized grid networks. 53

3.8 Plot of D_{avg} with respect to different $LL_{MaxLen2D}$ values in various sized grid networks. 54

4.1 Addition of six LLs (I to VI) to a 40-node STN to minimize the APL. . 58

4.2 Three cases to identify the locations of anchor nodes. 62

4.3 Objective function $P(p_1,\ p_2)$ in the constraint set $p_1 < p_2$. 64

4.4 Simulated observations on the locations of anchor nodes. 66

4.5 Path length contributions from various parts of a string network. 66

4.6 Identifying influence of the anchor nodes. 68

5.1 Greedy near-optimal decision based LL addition in an STN. 72

5.2 LL addition with SDLA algorithm in a string topology network. 73

5.3 Determination of the span distance in a string topology network. 75

5.4 Observations of APL variations and relative % deviations after LL addition with various LL addition strategies in STNs. 76

6.1 Three cases to identify the location of an optimal LL in a string topology WSN. 85

6.2 Two scenarios to add an optimal LL. 86

6.3 Tradeoff curves of APLB and the transmission power for M=1. 89

6.4 Tradeoff curves of APLB and transmission power for M=2. 92

7.1 Determination of min-cut between a source node and a destination node pair. 96

7.2 Modifying max-flow capacity with a new link addition. 98

7.3 MaxCap based LL addition in three different road networks. 101

7.4 ACES based LL addition in three different road networks. 103

7.5 ANFC performance of various arbitrary networks with fixed flow capacity. 105

7.6 ANFC performance of various arbitrary networks with variable flow capacity. 106

7.7 APL performance of various arbitrary networks with fixed or variable flow capacity. 107

7.8 ANFC performance of six real-world road networks with fixed flow capacity. 109

7.9 LTP performance of six real-world road networks with fixed flow capacity. 110

7.10 ANFC performance of six real-world road networks with variable flow capacity. 111

7.11 Variable flow capacity LL road-type distribution with ACES in six real-world road networks. 112

7.12 LTP performance of six real-world road networks with variable flow capacity. 112

7.13 APL performance of six real-world road networks with fixed or variable flow capacity. 113

List of Algorithms

3.1 Random LL Addition 37

3.2 Greedy Optimal LL Addition 42

3.3 Greedy Near-Optimal LL Addition 43

5.1 Sequential Deterministic LL Addition 74

7.1 Maximum Flow Capacity Algorithm 99

7.2 Average Flow Capacity Enhancement using Small-world Characteristics 102

Acronyms

ACC	Average Clustering Coefficient
ACES	Average flow Capacity Enhancement using Small-world characteristics
AND	Average Neighbor Degree
ANFC	Average Network Flow Capacity
APL	Average Path Length
APLB	Average Path Length to the Base Station
APSP	All Pair Shortest Path
BA	Barabási-Albert model
BC	Betweenness Centrality
BS	Base Station
CC	Closeness Centrality
DC	Degree Centrality
DN	Destination Node
EHD	End-to-end Hop Distance
ER	Erdös-Rényi
HND	Highest Neighbor Degree
LL	Long-ranged Link
LRA	Long-Ranged link Affinity
LTP	Length-Type Product
MaxBC	Maximum Betweenness Centrality
MaxCap	Maximum flow Capacity
MaxCC	Maximum Closeness Centrality
NFC	Network Flow Capacity
NL	Normal Link
PL	Path Length
SDLA	Sequential Deterministic LL Addition

SN	Source Node
STN	String Topology Network
SW	Small-World
SWSTN	Small-World String Topology Network
SWWSN	Small-World Wireless Sensor Network
WSN	Wireless Sensor Network

Notation

$\mathbb{R}$	Set of real numbers
$\mathcal{G}$	An arbitrary graph
$\mathcal{V}$	Set of vertices in a graph
$\mathcal{E}$	Set of edges in a graph
$\mathcal{C}$	Flow capacity
N	Total number of nodes in a network
$d_{(i,\ j)}$	Shortest path distance between nodes i and j
g_{jk}	Total number of shortest paths between nodes j and k
$g_{jk}(i)$	Number of shortest paths from node j to k that pass through node i
$BC(i)$	Betweenness centrality of node i
$CC(i)$	Closeness centrality of node i
$D(\mathcal{G})$	Diameter of graph $\mathcal{G}$
$DC(i)$	Degree centrality of node i
$\mathcal{O}(\cdot)$	Big O growth function
$//$	Comment line in an algorithm

Chapter 1

Introduction

Complex networks are networks that have complex and irregular connectivity patterns. Unlike regular networks that have clear motifs on the organization of nodes and edges, complex networks are challenging to understand and characterize. The study of complex networks is important due to the fact that complex networks appear in many aspects of our life, including, for example, biological networks, molecular networks, social networks, transportation networks, electric power grids, communication networks, and the Internet. In fact, most physical and biological systems can behave as a complex network of sub-systems and connectivity between them. Therefore, complex network modeling spans a large number of natural as well as man-made networks. Study of complex networks can in fact help in the study of many of those physical systems.

In previous study, scientists observed evolution and corresponding characterization of many real-world networks that are growing in the context of network size. However, there exists no such literature on the characterization of finite sized complex networks where the network evolution can be realized by addition of new links. This thesis investigates the evolution and characterization of such finite sized complex networks where new links are introduced based on the optimization of certain network metrics such as average path length, centrality metrics, and average network flow capacity.

This chapter explains the key motivation behind carrying out this research work which is followed by the major contributions of the thesis. Moreover, outline of the rest of the thesis is also provided at the end of this chapter.

1.1 Motivation and Objectives

This thesis is an attempt to answer some of the open questions exist in the context of the evolution in the finite sized complex networks. It can be seen that the evolution of

finite sized complex networks is not discussed much in the existing literature. Finite sized complex networks can be realized in many real-world examples, such as the network of relationships within a closed society, the airport network of certain country, the transportation network of a city, and the computer network of an organization [1]. To the best of our knowledge, this is the first attempt to study the evolution of such finite sized complex networks. Note that the evolution of finite sized network is not in the number of nodes, as considered in existing complex network literature, but in the number of links. In this thesis, we consider networks where new links[1] are added to an existing finite sized regular network, such as a string topology network (STN) or a grid topology network.

Many real-world regular networks achieve small-world characteristics [2, 3, 4] with a few long distance connections. By contrast, scale-free networks [5, 6, 7] that are abundantly found in nature, follow small-world characteristics along with the power-law degree distribution. Barabási et al. [5, 6] first explained the existence of scale-free characteristics that can be observed in many real-world networks. The scale-free characteristics can be achieved in a growing network when a newly introduced node preferentially attaches to a few existing nodes, decided by the fitness of those nodes to attract new connections. The process of the preferential attachment assumed to be a pure random phenomenon [8]. Papadopoulos et al. [9], conversely, observed that real-world network evolution is not based on pure luck [10], instead, scale-free characteristics in real-world networks can be realized, by optimization of the product of *similarity and popularity scores* of a network. In this thesis, we find a sequence of greedy optimal/near-optimal decisions can also contribute to the evolution of scale-free networks.

Therefore, in natural world, a sequence of greedy optimal/near-optimal decision making for long-ranged link (LL) addition can result in the transformation of a small-world network to a scale-free network [11]. By contrast, non-greedy LL addition does not result in a scale-free network. Our finding of gradual transformation of a small-world network to a scale-free network justifies that reason [5], optimization [9], or greed [11] plays key role behind the transformation of many real-world networks (e.g., the Internet and world-wide web (WWW), grid computing network, protein network in human body, ecological

[1]While creating a link in an existing network topology, the newly added link is named as *a long-ranged link* (LL) or simply, *a new link*. In this chapter, and rest of the thesis, LL and new link are used interchangeably.

interaction and food web, citation or social web networks, hubs in air traffic networks, and supply chain management networks) into scale-free networks. Hence, reason wins over luck in nature's scale-free network formation.

In this thesis, we consider addition of deterministic LLs in a finite sized STN to transform the network to a small-world network. We observe that addition of deterministic LLs, based on the optimization of certain network metrics such as average path length (APL), gradually transforms a finite sized regular network to a scale-free networks by introducing a few hub nodes where one end of most of the LLs are incident. Further, we observe that there are certain unique structural features when APL-optimal STNs are concerned. Note that APL-optimal finite sized complex networks are beneficial in many real-world deployments such as minimizing end-to-end hop distances, improving transmission delay, and keeping a predefined quality of service in network operation. Further, we also apply our observations to add optimal LLs (single as well as two) in various sized string topology wireless sensor networks. At last, we propose a novel LL addition technique to maximize average network flow capacity of a weighted complex network. In the next section, we detail our thesis contributions in what follows.

1.2 Major Contributions of the Thesis

Major contributions of this thesis are as follows:

1. We find that *greedy optimal/near-optimal decision* making,[2] based on certain network metrics such as APL, is one of the major reasons for the evolution of many real-world complex networks. The greedy optimal/near-optimal decision incorporates small-world characteristics; and along with the phenomenon of the long-ranged link affinity (LRA), a regular network can be transformed to a scale-free network with the formation of a few hub nodes.

2. While experimenting with the greedy near-optimal decision based LL addition in a finite sized STN, we discover that the location of the first LL always uniquely finds $0.2N^{th}$ and $0.8N^{th}$ nodes in an N-node network. We call the unique nodes as

[2]Note that we restrict most of our studies to the greedy near-optimal decision because the greedy optimal decision based LL addition is highly time complex, and not suitable for real-time applications.

the *anchor nodes*. We also analytically find fractional locations of the anchor nodes at 0.2071 and 0.7929 that validate our simulation observations on the evolution of various sized string topology network.

3. We also consider restriction on the length of an LL, known as constrained LL, and study the evolution of a finite sized complex network. The term *constrained LL* means that the creation of an LL obeys certain rules at the time of deployment in a network. We observe that, in a fixed sized complex network with the length constrained LL addition, there is a visible transition in the following manner: regular network $\rightarrow$ small-world network $\rightarrow$ scale-free network with the truncated degree distribution[3] $\rightarrow$ fully connected network.

4. Based on the simulation and analytical observations on a finite sized STN, we propose a heuristic strategy, sequential deterministic LL addition (SDLA), to efficiently transform a string topology network to an APL-optimal small-world network. SDLA adds k LLs in an N-node network only in $\mathcal{O}(k \times N)$ time compared to the greedy near-optimal LL addition strategy which takes $\mathcal{O}(k \times N^4 \log N)$ time.

5. We then analytically determine locations of a single and two optimal LLs, to optimize APL to the base station (APLB) value, in a string topology wireless sensor network (WSN). We also consider transmission power while optimizing the network APLB value. Our analytical observations reveal that one end of single as well as two LLs always connect to the BS which is assumed to be positioned at one end of the string WSN. The analytical findings also match with the simulation and approximate observations. The transformed small-world WSNs (SWWSNs) can be beneficial reducing transmission delay, enhancing network reliability and robustness, minimizing transmission packet loss, improving routing capability, and enhancing longevity of energy-constrained sensor nodes.

6. Finally, we apply small-world characteristics in order to enhance average network flow capacity (ANFC) of weighted undirected networks, such as real-world road

[3] *Truncated degree distribution* is a conditional distribution imposed by certain restriction. Details of the truncated distribution can be found in Chapter 3.

networks. We propose an exhaustive search based LL addition algorithm, maximum flow capacity (MaxCap), which deterministically maximizes the ANFC value based on the maximum flow between node pairs in a weighted undirected network. Based on the observations from MaxCap, we construct a new LL addition heuristic, average flow capacity enhancement using small-world characteristics (ACES), that significantly enhances ANFC, and the length-type product (LTP) of a network.

In conclusion, this thesis aims at understanding the evolution of finite sized complex networks (based on fixed sized string and grid network topologies) and applies the observations in designing efficient real-world networks to achieve certain performance objectives.

1.3 Thesis Outline

The remaining thesis is structured as follows:

In **Chapter 2**, a review of important performance metrics, that are extensively used to study finite sized complex network topologies, are carried out. We also introduce a few complex network models, such as regular networks, random networks, small-world networks, and scale-free networks, along with their characteristics.

In **Chapter 3**, greedy optimal/near-optimal decision based LL addition is carried out to transform a finite sized string network to an APL-optimal small-world network. We find that, while adding LLs with optimal/near-optimal decision, the string network gradually evolves to an APL-optimal scale-free network. We identify that the first LL, based on the APL-optimal decision in a string topology, always finds a fixed fractional locations, referred as the *anchor nodes*.

Further, we put restrictions on the length of added LLs to study various transition phases in the evolution of fixed sized complex networks. Based on the simulation observations on the finite sized string and grid network topologies, we observe that a fixed sized network gradually evolves as follows: a regular network $\rightarrow$ a small-world network $\rightarrow$ a scale-free network with truncated degree distribution $\rightarrow$ a fully connected network.

In **Chapter 4**, we analytically find the fractional locations of the anchor nodes are

at 0.2071 and 0.7929, where the first APL-optimal LL in a finite sized STN is added. The analytical result also validate our simulation observations based on the greedy optimal/near-optimal LL addition. We also study the importance of identification of anchor nodes to design real-world networks.

In **Chapter 5**, a heuristic approach, sequential deterministic LL addition (SDLA) is developed, based on the simulation and analytical observations on finite sized unweighted STNs. SDLA can efficiently transform a regular STN to an APL-optimal small-world network. The heuristic can efficiently deploy a few LLs in disaster response and other ad hoc network deployment scenarios.

In **Chapter 6**, we analytically determine optimal locations of a single and two LLs, in order to optimize the APLB values in a string topology WSN. We observe that one end of a single optimal LL or two optimal LLs always connect to the base station (BS) which is assumed to be situated at one end of a string topology WSN. We also incorporate transmission power expended while creating an LL and observe that, due to the tradeoff between the transmission power and the length of an LL, optimal locations of new LLs are varied to account for the tradeoff constraints.

In **Chapter 7**, we study a set of arbitrary networks, such as STNs, grid networks, and random networks, as well as a few real-world networks, such as road networks where small-world characteristics are incorporated by creating a handful of LLs to enhance the ANFC values. We also consider addition of a few LLs deterministically to weighted undirected road networks in order to maximize the ANFC. Based on the observations from LL addition, with the exhaustive search based strategy, we propose a heuristic approach that can enhance the ANFC of a weighted network along with the LTP value.

In **Chapter 8**, we conclude our thesis with a few possible future research directions.

1.4 Summary

Complex networks, abundantly found in natural and man-made networks, are abstract graphs with non-trivial topological features. Most of the existing literature focused on understanding the evolution of real-world growing complex networks. However, there

are very limited literature in the context of the evolution of finite sized complex networks. In this thesis, we work toward gaining insights on the evolution of fixed sized complex networks. In this chapter, we discussed key motivation of our thesis work, major contributions of the thesis, and the organization of rest of the thesis.

Chapter 2

Introduction to Complex Networks

A large number of physical systems in nature and technology are formed by a collection of highly connected dynamical entities that can be modeled, represented, and characterized as complex networks.[1] The term *complex networks* is coined from the fact that all such physical systems have non-trivial topological features which cannot be thoroughly studied and characterize with the conventional graph theoretical models. One such example of a complex network is an electrical power-grid network formed by power generation sources, distribution lines, switches that control the transmission of electricity, and consumer equipments. Another popular example of a complex network is a computer network such as the Internet that consists of servers, clients, switches, and routers interconnected by several communication links such as optical fiber, co-axial cables, Ethernet cables, wireless links, and satellite links. Furthermore, natural networks such as biological networks, food-web networks, protein-protein interaction networks, diseasome networks [12], and ecological networks to human-made networks such as the author citation networks [13, 14], the world wide web (WWW) [15], transportation networks, and mobile call networks [3, 7, 16, 17, 18, 19] are a few examples of complex networks.

This chapter discusses a few metrics that can be used to analyze characteristics of large sized complex networks. Moreover, we study a few complex network models such as regular, random, small-world, and scale-free networks which are heavily referenced to characterize and study many real-world networks.

2.1 Real-World Complex Networks

An abstract model of any complex physical system gives rise to a complex network model. Networks obtained by modeling complex systems result in a complex interconnection of nodes (vertices) by links (edges or arcs). Such models of complex physical systems are

[1] In this chapter, and rest of the thesis, we use the terms *network* and *graph* interchangeably.

considered as complex networks. In the following, we discuss three real-world complex networks: (i) an author citation network, (ii) the autonomous systems in the Internet, and (iii) an air traffic network. Note that all the real-world networks discussed in this section are realized with Gephi 0.9.1 graph visualization and analysis tool where various communities are depicted in different colors.[2]

2.1.1 The Author Citation Networks

A co-authorship network is an example of a real-world complex network. In a co-authorship network, nodes are researchers or authors, whereas links among nodes are considered when a research paper is co-authored by two researchers. The size of a circle is based on how many times an author has co-authored papers with other researchers. It also identifies a key person on a certain specialization. Figure 2.1 has only a few nodes with many connections (big circles), which act as hub nodes. Most of the nodes are attached with only a small number of links. Therefore, in the figure, big circles are very influential researchers in the particular field (here, we consider networking) where many researchers have co-authored with them. It can also be noticed from Figure 2.1 that there exist a couple of scientific communities, in different specializations, on the broader research areas of network theory.

2.1.2 The Autonomous Systems in the Internet

The Internet is an example of a technological complex network (see Figure 2.2). Figure 2.2 shows a segment of the Internet that is too large to visualize in the page of this thesis. In this network, a node is an autonomous system (AS) and interconnection between two ASs is denoted by a link. An AS is a segment of the Internet under the control of an autonomous administration. For example, an organization's entire network can typically be considered as a single AS. Similar to the author citation network, a few nodes in the Internet have millions of connections, whereas the rest of the nodes are associated with only a few neighbors.

[2]Communities in real-world networks can be realized with the modularity score. Note that a high modularity score shows better internal community structure and helps in deciding the subnetwork compartmentalization.

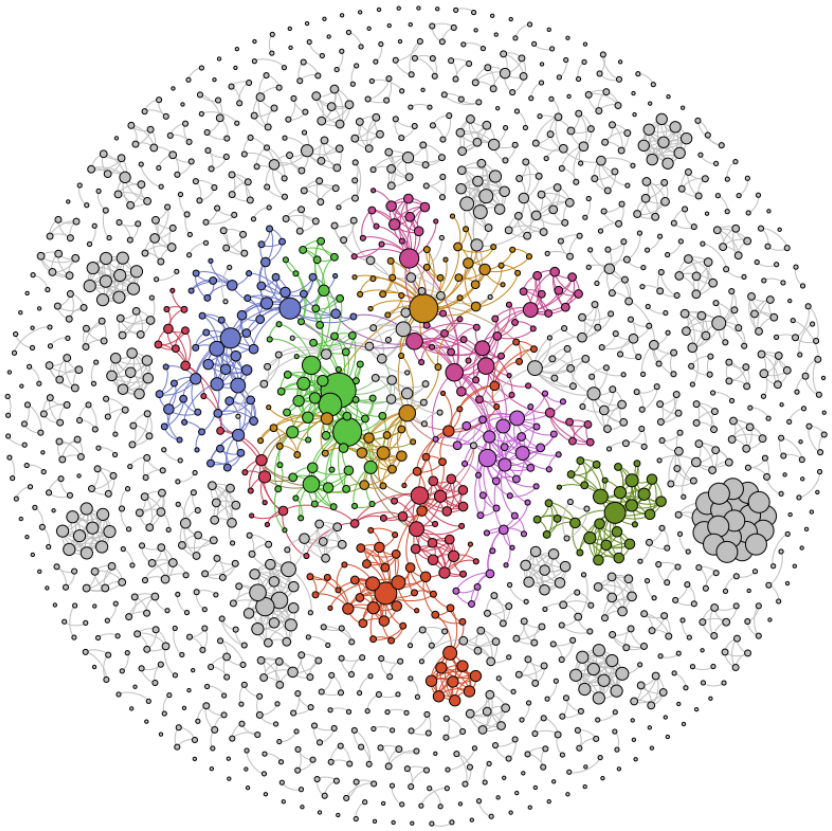


Figure 2.1: An example author citation network [20]. Data is drawn from a coauthorship network on the scientific work on network theory. Here, a node is an author of a paper, and a link exists between two authors if they co-authored a paper. The graph is generated with Gephi 0.9.1, and the network layout is Fruchterman-Reingold.

2.1.3 The Air Traffic Networks

Another example of real-world complex network is air traffic network, as shown in Figure 2.3. Here, nodes in the network represent airports, and if a flight connects two airports, a link is connected between the two. In the figure, it is observed that a few airports are connected to a large number of airports (denoted as big black circle) that act as hub nodes. One of the key purposes of a hub node is to create the shortest routes to economically reach the most locations through the air traffic network.

In order to study the behavior of complex networks, a few important complex network metrics are presented in the following.

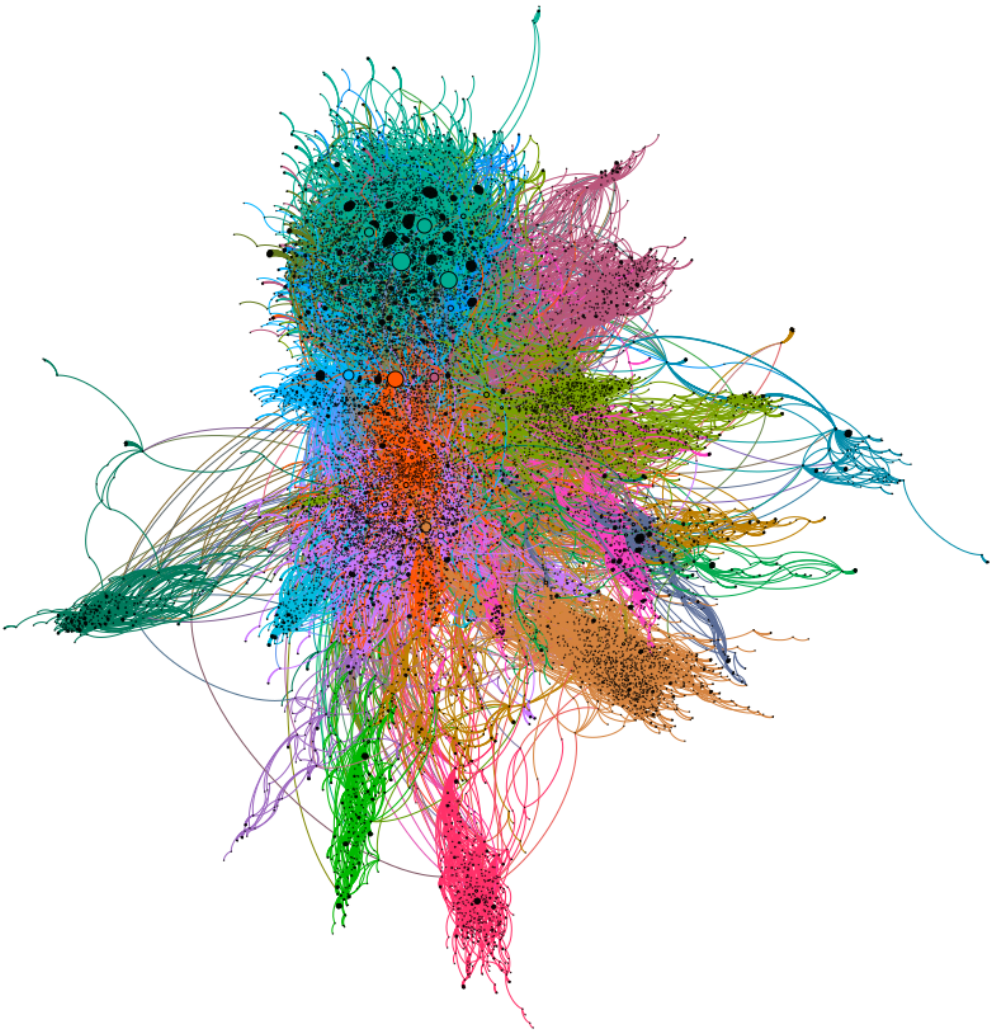


Figure 2.2: The network of autonomous systems (ASs) in the Internet [21] with 22,963 nodes and 48,434 links. A node in the network represents an AS and an edge represents interconnection between two ASs. The graph is generated with Gephi 0.9.1 and the network layout is ForceAtlas 2.

2.2 Complex Network Metrics

As the real-world complex networks are always evolving with time, microscopic study of the networks, to understand their characteristics, is not a feasible solution. In order to thoroughly characterize such large sized complex networks, one needs to view the networks macroscopically. There exist many popular metrics that can be used to measure the macroscopic properties of complex networks. Examples of such metrics are (i) average nodal degree (AND), (ii) average clustering coefficient (ACC), (iii) average path length (APL), (iv) network diameter, (v) degree distribution, and (vi) centrality metrics.

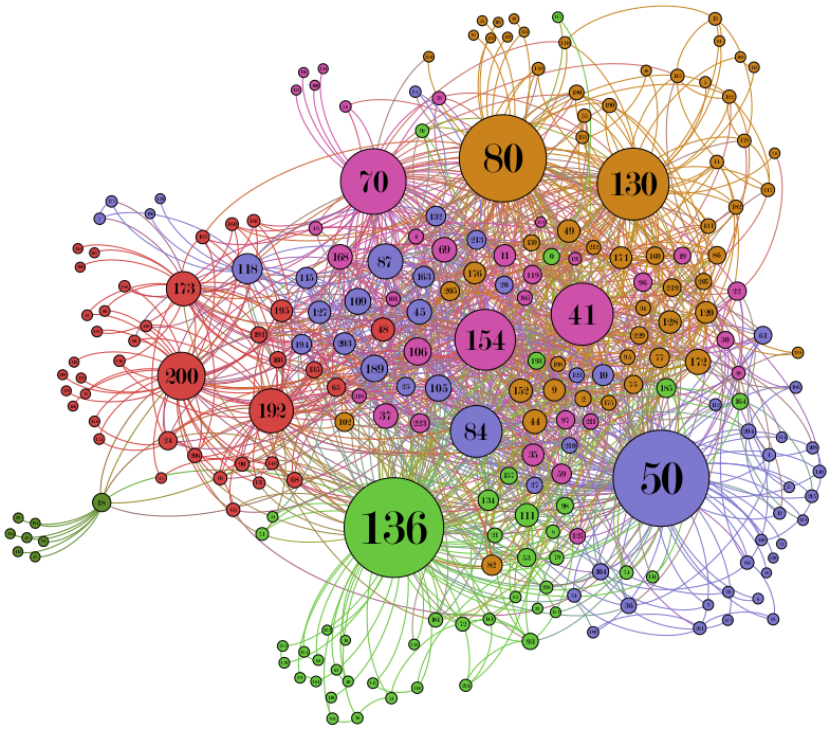


Figure 2.3: An air traffic network [22] with 235 airports (nodes) in various part of the world and 1297 flights (edges) that run between the airports. The graph is generated with Gephi 0.9.1, and the network layout is ForceAtlas 2.

2.2.1 Average Nodal Degree

The AND of a graph $\mathcal{G}$ with N nodes can be defined as

$$\text{AND}(\mathcal{G}) = \frac{1}{N}\sum_{i=1}^{N} d_i, \tag{2.1}$$

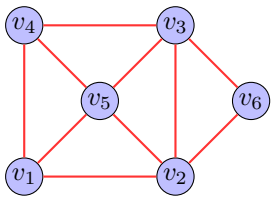
where d_i is the degree of node i. For example, the AND value of the graph corresponding to Figure 2.4(a) can be estimated as $\frac{1}{6} \times [(3 \times 4) + (2 \times 3) + (1 \times 2)] = \frac{20}{6} = 3.33$. Note that, in Figure 2.4(a), each link is assumed to be of unit weight.

2.2.2 Average Clustering Coefficient

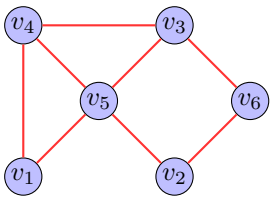
The ACC value reveals local connectivity property of a network graph. ACC is measured by taking summation of the clustering coefficient for each node averaged over the number of nodes in the network. Hence, ACC for a network consisting of N number of nodes is calculated as

$$ACC = \frac{1}{N} \times \sum_{\substack{i \in N \\ \forall m_i}} \frac{2 \times e_{pq}}{n_{m_i} \times (n_{m_i} - 1)}, \tag{2.2}$$

where $v_p,\ v_q \in V_{m_i},\ e_{pq} \in E$. Note that, in Equation (2.2), m_i is the number of one-hop neighbor nodes (i.e., immediate neighbors) for i^{th} node in the network, and e_{pq} is the one hop connection between neighbors p and q (i.e., v_p and v_q). Therefore, the summation is taken as the total number of neighbor connections among n_{m_i} neighbors in the network over the maximum possible connections among the neighbors. In Equation (2.2), 2 is included due to the bidirectional (for directional link, possible number of links can be formed among n_{m_i} neighbors are $[n_{m_i} \times (n_{m_i} - 1)]$) nature of the links. However, CC value for a node with only one neighbor is considered to be zero. Figures 2.4(a) and (b) depict example graphs to determine CC of a node.



(a) $\mathcal{G}_1$.



(b) $\mathcal{G}_2$.

Figure 2.4: Example graphs for ACC calculation.

In Figure 2.4(a), node v_5 has four neighbors. Therefore, to calculate the clustering coefficient for node v_5, it can be found that the possible neighbor connections, among the four neighbors, is six. However, only four connections exist among the neighbor nodes. Therefore, the clustering coefficient value for node v_5 shown in Figure 2.4(a) is $\frac{4}{6}$. Similarly, for graph $\mathcal{G}_2$ of Figure 2.4(b), the clustering coefficient value of v_5 is $\frac{2}{6}$. The clustering coefficient for individual nodes can be averaged to obtain a network's ACC. For the graph $\mathcal{G}_1$, the clustering coefficient for each node is as follows: $v_1 = \frac{2}{3}$, $v_2 = \frac{3}{6}$, $v_3 = \frac{3}{6}$, $v_4 = \frac{2}{3}$, $v_5 = \frac{4}{6}$, and $v_6 = 1$. The ACC of the graph $\mathcal{G}_1$ can be estimated as $\frac{2}{3}$. Similarly, for graph $\mathcal{G}_2$ of Figure 2.4(b), the value of the ACC is $\frac{2}{9}$.

The ACC value in a *regular network* is typically low to moderate, as the neighbor nodes are well connected. However, immediate neighbor nodes in a *random network* may not always be connected. Therefore, ACC for a random network is typically low.

2.2.3 Average Path Length

APL, which is a global network property, can be measured by the mean shortest path distance or geodesic between two nodes averaged over all the nodes in the network graph. The APL value of a network consisting of N nodes can be calculated as

$$APL = \frac{2}{N \times (N-1)} \sum_{i \neq j} d_{(i,\, j)}, \tag{2.3}$$

where $d_{(i,\, j)}$ is the shortest hop distance between nodes i and j. Therefore, APL is calculated as the summation of all shortest hop distances in a network averaged over all possible connections in the network (in Equation (2.3), due to the bidirectional links in the network, 2 is included).

In order to estimate the APL value of the graph $\mathcal{G}_2$ in Figure 2.4(b), at first all shortest path distances from each node has to be evaluated. Then all such shortest path distances, from all nodes, are added and divided by the maximum number of link possibilities of the network. Therefore, the shortest path distances from node v_1 are as follows: $v_2 : 2$, $v_3 : 2$, $v_4 : 1$, $v_5 : 1$, and $v_6 : 3$. Hence, the total shortest path distance from node v_1 to all other node is 9. The total shortest path distance from all nodes to all other nodes is calculated to be $(9 + 8 + 7 + 7 + 6 + 9) = 46$. Therefore, the APL value of the graph $\mathcal{G}_2$ in Figure 2.4(b) can be estimated as $\frac{46}{(6\times5)} = \frac{46}{30} \simeq 1.53$. Likewise, the APL value of the graph $\mathcal{G}_1$ in Figure 2.4(a) can be estimated as approximately 1.33.

2.2.4 Network Diameter

The diameter of a graph $\mathcal{G}$ is equal to the largest shortest path between any node pair in the graph. It is represented by $D(\mathcal{G})$. Let $d_{(i,\, j)}$ represents the shortest path distance between nodes i and j in a graph, then the network diameter can be expressed as

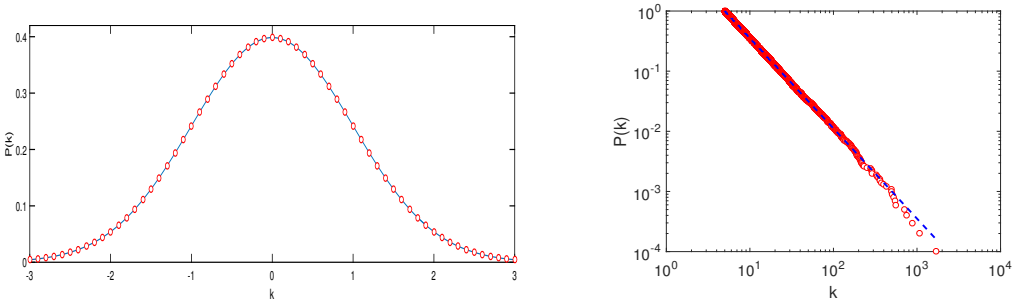
$$D(\mathcal{G}) = \max_{\forall i,j} \{d_{(i,\, j)}\}. \tag{2.4}$$

In other words, $d_{(i,\, j)}$ represents the shortest path distance between nodes i and j in a network, and the diameter represents the maximum of all shortest path distances.

The network diameters of the graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ of Figures 2.4(a) and (b) are 2 and 3, respectively.

2.2.5 Degree Distribution

Degree distribution of a network reflects overall connectivity profile of a network. If k is the degree of a node, $P(k)$ measures the probability of a node with degree k. The degree distribution is generally plotted by taking the normalized $P(k)$ values. Figure 2.5(a) shows a normal (Gaussian) distribution with the mean value at 0.



(a) **(b)**

Figure 2.5: Examples of (a) normal degree distribution with zero mean value and (b) power-law degree distribution (in $\log - \log$ scale) with slope = 2.5. Note that, here k represents degree of a node and $P(k)$ denotes probability of finding nodes with degree k.

The degree distributions of many real-world complex networks do not follow normal distribution, instead, the real-world networks follow power-law distribution and classified as *scale-free networks* (see Section 2.3.4 for a detailed discussion on scale-free networks). In power-law distribution, the gradient of the distribution follows the relation $P(k) \sim k^{-\gamma}$ or $P(k) = rk^{-\gamma}$, where r is a constant and $\gamma \in \mathbb{R}$. By taking logarithm at both sides of the expression, we get $\log(P(k)) = \log(r) - \gamma \log(k)$. Thus, the power-law distribution has a negative gradient of γ with a Y-axis cut at $\log(r)$. Figure 2.5(b) shows an example power-law curve on a $\log - \log$ scale with $\gamma = 2.5$.

2.2.6 Centrality Metrics

Measure of importance or centrality is fundamental in understanding the structural and dynamic properties of complex networks. Measuring centrality of a node in a network is quantifying the importance of that node in the network. This quantification is carried out based on various features, such as number of neighbors, ability of a node to quickly

communicate with other nodes, role of a node in the flow control between other nodes, and the influence of neighbors on a node. A number of centrality measures exist in the literature, but one basic question arises: Which centrality measure is the best? The answer to this question depends on the application at hand. One centrality measure may work well in a particular application, however, it may fail in another. In the following, some of the popular centrality measures are discussed.

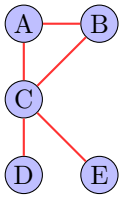
Degree Centrality

The degree centrality (DC) is the simplest measure of centrality. DC of a node can be defined as the sum of the edge weights incident on that node. DC of a node i in any network can be calculated by the following equation:

$$DC(i) = \sum_{j} e_{ij}, \qquad \forall e_{ij} \in E. \tag{2.5}$$

For any N-node network, normalized value of DC can be realized by comparing centrality of a node in that network with respect to the central node of a star network consists of N nodes (as the center node of a star network has the highest degree, i.e., $N-1$). Thus, normalized DC ($DC'(i)$) of any network can be achieved by taking the ratio of degree of the i^{th} node to the degree of the central node of a star network, as given in the following equation:

$$DC'(i) = \frac{\sum_j e_{ij}}{N-1}, \qquad \forall e_{ij} \in E. \tag{2.6}$$



(a) An example network.

$$\begin{bmatrix} & A & B & C & D & E \\ A & 0 & 1 & 1 & 0 & 0 \\ B & 1 & 0 & 1 & 0 & 0 \\ C & 1 & 1 & 0 & 1 & 1 \\ D & 0 & 0 & 1 & 0 & 0 \\ E & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

(b) Adjacency matrix.

Figure 2.6: An example network and its adjacency matrix.

Figure 2.6(a) shows a sample unweighted network and corresponding adjacency ma-

Table 2.1
Degree centrality

Node	DC	DC′
A	2	2/4
B	2	2/4
C	4	1
D	1	1/4
E	1	1/4

trix[3] is mentioned in Figure 2.6(b). The DC scores of the nodes are listed in Table 2.1. It can be seen that the DC score of node C is the highest, implying that node C is the most central node (according to the degree based centrality measure).

Closeness Centrality

The closeness centrality (CC) measures how close a node is to other nodes in a network. Nodes that are close in a network can interact with their neighbor nodes very quickly. CC also measures the importance of a node in spreading information to other nodes in a network. CC of i^{th} node (i.e., $CC(i)$) in an N-node network can be measured by the following equation:

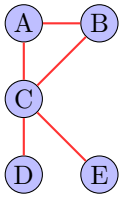
$$CC(i) = \frac{1}{\sum_{j=1}^{N} d_{(i,\,j)}}, \tag{2.7}$$

where $d_{(i,\,j)}$ is the length of the shortest path between nodes i and j. To get the normalized value (i.e., $CC'(i)$) with respect to a star topology network, the following equation is used [23]:

$$CC'(i) = \frac{N-1}{\sum_{j=1}^{N} d_{(i,\,j)}}. \tag{2.8}$$

Figure 2.6(a) shows a sample unweighted network and corresponding cost matrix[4] is depicted in Figure 2.6(b). CC scores of the nodes are listed in Table 2.2. It can be seen that, according to CC measures, node C is the most central (important) node in the network. Also note that node C receives a maximum CC score of 1, because it is

[3] *Adjacency matrix* represents whether a link, between a node pair, is present (by 1) or not (by 0).
[4] *Cost matrix* represents shortest path distance between a node pair in a network.



(a) An example network.

$$\begin{bmatrix} & A & B & C & D & E \\ A & 0 & 1 & 1 & 2 & 2 \\ B & 1 & 0 & 1 & 2 & 2 \\ C & 1 & 1 & 0 & 1 & 1 \\ D & 2 & 2 & 1 & 0 & 2 \\ E & 2 & 2 & 1 & 2 & 0 \end{bmatrix}$$

(b) Cost matrix.

Figure 2.7: A sample network and its cost matrix.

Table 2.2
Closeness centrality

Node	$\sum_j d(i,\ j)$	CC	$CC^{'}$
A	6	1/6	4/6
B	6	1/6	4/6
C	4	1/4	1
D	7	1/7	4/7
E	7	1/7	4/7

connected to all other nodes in the network.

Betweenness Centrality

Communication between two non-adjacent nodes in a network can be achieved via multiple paths. The betweenness centrality (BC) measures the extent to which one node lies between the shortest paths of other nodes in the network. Therefore, BC measures the importance of one node in making long-distance communications. BC can be measured by calculating all possible shortest paths that pass through a particular node, as given by

$$BC(i) = \sum_{i \neq j \neq k} \frac{g_{jk}(i)}{g_{jk}}, \tag{2.9}$$

where g_{jk} is the total number of shortest paths from node j to k in an N-node network, and $g_{jk}(i)$ is the number of paths that pass through node i. To normalize the BC value for a node, Equation (2.10) [23] is used:

$$BC^{'}(i) = \frac{BC(i)}{[(N-1)(N-2)/2]}. \tag{2.10}$$

In the network shown in Figure 2.8, nodes 1, 2, 4, and 5 are not present in any of the shortest paths between any pairs of nodes in the network, thereby, result in score of 0 BC

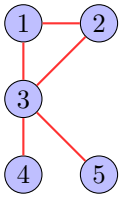


Figure 2.8: An example network to calculate betweenness centrality.

Table 2.3
Betweenness centrality

Node	BC	BC'
1	0	0
2	0	0
3	5	5/6
4	0	0
5	0	0

for these nodes. In contrast, node 3 is present in various shortest paths. The BC score for node 3 can be found as

$$\begin{aligned} BC(3) &= \frac{g_{12}(3)}{g_{12}} + \frac{g_{14}(3)}{g_{14}} + \frac{g_{15}(3)}{g_{15}} + \frac{g_{24}(3)}{g_{24}} + \frac{g_{25}(3)}{g_{25}} + \frac{g_{45}(3)}{g_{45}} \\ &= 0 + \frac{1}{1} + \frac{1}{1} + \frac{2}{2} + \frac{2}{2} + \frac{1}{1} = 5. \end{aligned}$$

Therefore,

$$BC'(3) = \frac{BC(3)}{(5-1)(5-2)/2} = \frac{5}{6}.$$

Graph Centrality

The graph centrality (i.e., GC_{metric}) is another important metric, which represents a compact view of the network characteristics. That is, graph centrality identifies network centralization on the basis of the node-level information. GC_{metric}, in the context of DC, CC, or BC, of an N-node network can be identified by the following equation:

$$GC_{metric} = \frac{\sum_{i=1}^{N}[GC_{metric}(x^{\star}) - GC_{metric}(x_i)]}{\max \sum_{i=1}^{N}[GC_{metric}(x^{\star}) - GC_{metric}(x_i)]}, \tag{2.11}$$

where $GC_{metric}(x_i)$ is the centrality of node i, $GC_{metric}(x^\star)$ is the largest value of node centrality in the N-node network, and *metric* can be any of the centrality metrics mentioned earlier. The denominator of Equation (2.11) identifies the maximum difference in the node centrality [23].

GC_{metric} measures the deviation of $GC_{metric}(x^\star)$ with respect to the remaining nodes in a network. The operating range of the graph centrality metric is $0 \leqslant GC_{metric} \leqslant 1$. Here, $GC_{metric} = 0$ indicates that all nodes in the network are of equal importance, whereas, $GC_{metric} = 1$ indicates that the node with the highest centrality value dominates the remaining nodes in the network. The graph centralities with DC (i.e., GC_{DC}), CC (i.e., GC_{CC}), and BC (i.e., GC_{BC}) of an N-node graph can be estimated with the following equations [23]:

$$GC_{DC} = \frac{\sum_{i=1}^{N}[GC_{DC}(x^\star) - GC_{DC}(x_i)]}{N^2 - 3N + 2}, \tag{2.12}$$

$$GC_{CC} = \frac{\sum_{i=1}^{N}[GC_{CC}(x^\star) - GC_{CC}(x_i)]}{(N^2 - 3N + 2)/(2N - 3)}, \tag{2.13}$$

$$\text{and} \quad GC_{BC} = \frac{\sum_{i=1}^{N}[GC_{BC}(x^\star) - GC_{BC}(x_i)]}{N^3 - 4N^2 + 5N - 2}. \tag{2.14}$$

2.3 Complex Network Models

Complex networks have complex and irregular patterns of connectivities among network nodes such that the ordinary graph theoretical approaches cannot be directly applied to understanding the characteristics of such networks. That is, the topological features of complex networks can be considered as non-trivial compared to regular networks. In many cases, the non-trivial connectivity pattern is exacerbated by the large size of the network in terms of the number of nodes and edges. When complex physical systems such as biological networks, social networks, technological networks, and the Internet are modeled as graphs, complex networks result [24, 25].

Existing real-world complex networks can be broadly classified as: (i) regular net-

works, (ii) random networks, (iii) small-world networks, and (iv) scale-free networks. In the following, a brief description of each category is provided.

2.3.1 Regular Networks

A node in a regular network is mostly connected to all of its immediate neighbors [26]. For example, each node in an r-regular network is connected to its r neighbors. Consider all possible graphs with N nodes each with degree r, and then select one of the graph models at random to get a random r-regular network [27]. An example of a 4-regular network is shown in Figure 2.9.

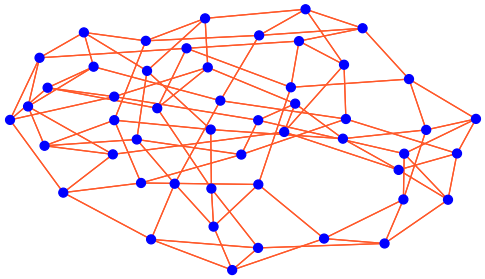


Figure 2.9: An example 4-regular network of 50 nodes.

A regular network has moderate to high value of ACC because most of its neighbors are also connected.[5] Thus, regular networks are robust against multiple link failures. However, the APL value is also high as it takes multiple hops to reach a distant destination node from a source node.

2.3.2 Random Networks

A random network can be evolved by randomly choosing a set of node pairs out of all possible node pairs in a network. To create a random network of N nodes, the following two approaches can be exercised:

- If the total number of links (e.g., M) are known, then the network model can be constructed based on the generating function $\mathcal{F}(N,\ M)$. That is, the random net-

[5] A string topology network or a grid topology network is a special case of regular network where the ACC value is 0.

work can be realized by creating M links randomly out of $\binom{N}{2}$ possibilities in an N-node network.

- Conversely, if the probability of the link creation (i.e., p), between a node pair is given, then a random network can be evolved by adding new links based on the link creation probability p. Therefore, the graph creation model can be expressed with the generating function $\mathcal{F}(N,\ p)$. This approach of random network creation is also popularly known as the *Erdös-Rényi (ER)* random network model [28, 29].

Erdös-Rényi Random Network Model

In an ER network, a node pair is connected if the presence of that link satisfies the link creation probability p. That is, while creating a link between a node pair, the link creation probability is compared with the expected link creation probability p and the link is added if the link creation probability is greater than or equal to p. An ER graph consisting of 50 nodes is depicted in Figure 2.10. The edges, in this figure, are randomly connected among node pairs with probability $p = 0.06$.

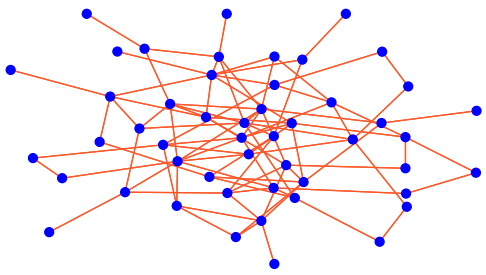


Figure 2.10: An example ER-network of 50 nodes with p = 0.06.

As shown in the figure, the immediate neighbors (i.e., nearby in terms of distance) may not always be connected in an ER-network. As a result, the ACC value is low for the ER network [30]. However, due to the presence of a few links between distant node pairs, APL value of an ER-network is lower compared to a regular network.

2.3.3 Small-World Networks

A small-world network lies between a regular network and a random network, and incorporates the best features from both the networks. In a regular network, most of the nodes

are well connected and hence, the ACC value is moderate to high. However, as multiple hops are required to reach a distant node from a source node, the APL value is also high in the context of a regular network. On the other hand, the ACC value is low as the neighbor nodes are not well connected in a random network. However, the APL value is also lower in the random network as there exist a few long distant connections.

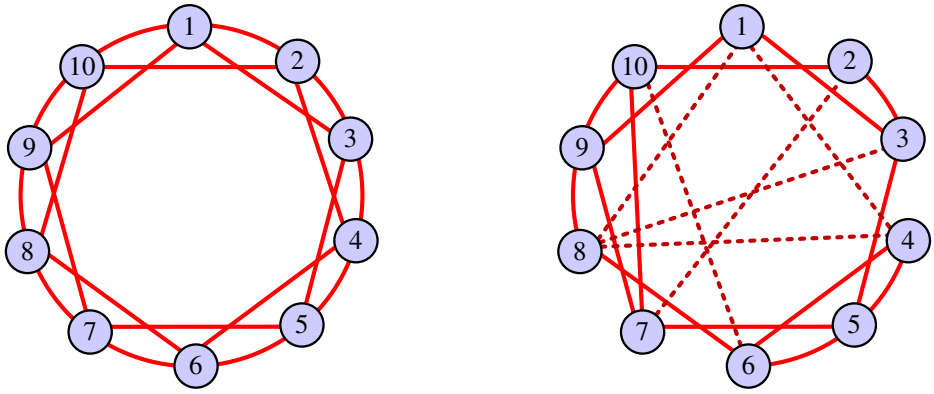
In a small-world network, most of the nodes are connected like a regular network topology and a small number of long-ranged links (LLs) are also present among distant node pairs. Therefore, small-world networks exhibit lower value of APL along with moderate value of ACC. A comparison of the three network topologies in terms of ACC and APL is depicted in Table 2.4.

Table 2.4
The table compares regular networks, small-world networks, and random networks based on the ACC and APL values. In a regular network, both the ACC and APL values are moderate to high. However, the APL value is low for random networks along with lower value of the ACC. Small-world networks inherit the best characteristics from regular as well as random networks and exhibit lower value of the APL with moderate value of the ACC. Asymptotic APL values are also provided in the following table.

Parameters	**Regular Networks**	**Small-world Networks**	**Random Networks**
ACC	Moderate	**Moderate**	Low
APL	High	**Low**	Low
Asymptotic APL Values	$\mathcal{O}(N)$	$\mathcal{O}(\log \mathbf{N})$	$\mathcal{O}(\log N)$

From Table 2.4, it can be observed that regular networks and random networks are the two extreme scenarios when network topologies are concerned.

A regular network can be transformed to a small-world network by either rewiring minimal number of existing normal links (NLs) or adding a few LLs. The key characteristics of a small-world network are lower values of the APL with low to moderate values of the ACC. Figure 2.11(a) shows an example of a 10-node 4-regular network with APL = 1.67 and ACC = 0.50. However, when very few existing NLs are rewired with the rewiring probability $p = 0.2$ (a detailed discussion on rewiring can be found later in this section), values of APL = 1.58 and ACC = 0.34 are changed in the resultant network, as shown in Figure 2.11(b).



(a) APL = 1.67, ACC = 0.50. **(b)** APL = 1.58, ACC = 0.34.

Figure 2.11: (a) An example 4-regular network. (b) A small number of NLs of (a) are rewired with $p = 0.2$. The resultant network transforms to a small-world network.

It can be seen that as the size of the network in Figure 2.11(a) is small, change in the overall APL value is not significant (only 6.59%) with rewiring probability of 0.2. As the network size increases, improvement in the APL value can be observed. Table 2.5 shows a few numerical observations in the context of different sized 4-regular networks (network sizes are 100, 200, 300, 400, and 500). Note that the data in Table 2.5 is generated by rewiring a few NLs with rewiring probability $p \in \{0.05,\ 0.1,\ 0.2,\ 0.5\}$.

From Table 2.5, it can be observed that as the network size increases, reduction in the APL value is also improved. For example, with rewiring probability $p = 0.05$ in a 100 node network, reduction of the APL value with respect to the regular network, is approximately 44.18%. Conversely, the ACC value does not decrease in the same manner compared to the improvement in the APL value. The decrease in the ACC value is approximately 10% when rewiring with $p = 0.05$ in a 100 node network is concerned.

Instead of rewiring the existing NLs, new LLs can also be added in a regular network to transform it to a small-world network. However, as time elapsed, with more number of LLs the network becomes a fully connected mesh. In the following, a few small-world network evolution models are discussed.

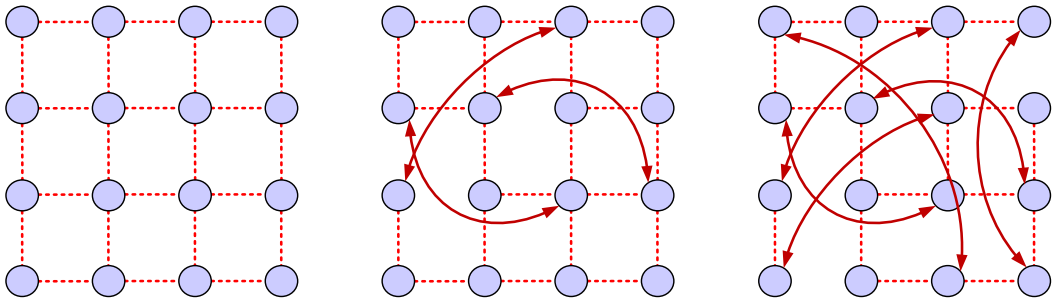
Rewiring of Existing Links

Rewiring is one of the mechanisms by which a regular network can be transformed to a small-world network. In rewiring, some of the existing NLs are rewired to other nodes

Table 2.5
The table shows data for transformation from various sized 4-regular networks (N) to small-world networks with different rewiring probabilities p. In this table, $N \in \{100,\ 200,\ 300,\ 400,\ 500\}$ and $p \in \{0.05,\ 0.10,\ 0.20,\ 0.30,\ 0.40,\ 0.50\}$. The ACC value is always 0.50 before the rewiring operation. It can be seen that APL value is drastically reduced with increasing p values. However, there is also a constant decrease in the ACC value. The table also provides the percentage APL reductions with respect to the APL values prior to rewiring.

No. of Nodes	Rewiring Probability (p)	APL before Rewiring	APL after Rewiring	Percentage Reduction in APL	ACC after Rewiring
100	0.05	12.88	7.19	44.18	0.45
	0.10		4.55	64.67	0.32
	0.20		4.01	68.87	0.24
	0.50		3.50	72.83	0.07
200	0.05	25.38	7.40	70.84	0.43
	0.10		6.04	76.20	0.36
	0.20		4.91	80.65	0.25
	0.50		4.17	83.57	0.09
300	0.05	37.88	9.77	74.21	0.44
	0.10		6.57	82.66	0.36
	0.20		5.20	86.27	0.24
	0.50		4.55	87.99	0.09
400	0.05	50.38	9.29	81.56	0.43
	0.10		7.58	84.95	0.38
	0.20		5.93	88.23	0.29
	0.50		4.73	90.61	0.06
500	0.05	62.88	11.68	81.42	0.45
	0.10		7.64	87.85	0.37
	0.20		5.86	90.68	0.25
	0.50		5.00	92.05	0.08

based on certain probability p, where $0 \leqslant p \leqslant 1$ [3]. In the process of rewiring, a regular network can be transformed to a completely random network. During the transformation from regular networks to random networks, small-world networks can be observed when the rewiring probability is lower. This situation is depicted in Figure 2.12. In Figure 2.12(a), a regular grid network is shown with the rewiring probability $p = 0$. AND of the regular network is moderate, whereas the APL value is large because the end-to-end hop distances between the distant node pairs are more in Figure 2.12(a).



(a) $p = 0$. **(b)** $0 < p < 1$. **(c)** $p \to 1$.

Figure 2.12: Rewiring in a regular grid lattice. (a) A regular grid with the rewiring probability $p = 0$. (b) The grid is transformed to a small-world network with the rewiring probability $0 < p < 1$. (c) The grid is transformed to a random network with the rewiring probability $p \to 1$. Here, the dashed lines represent the NLs and the bidirectional links represent the LLs.

When some NLs are removed from one end and then are reconnected with probability $0 < p < 1$ to certain distant nodes (the bidirectional solid line in Figure 2.12(b)) as LLs, the APL value of the network is reduced. Thus, in the process of rewiring a small number of existing NLs, lower value of APL can be attained. However, the ACC value is kept nearly unchanged compared to the regular network. When NLs are rewired with the rewiring probability $p \to 1$, as shown in Figure 2.12(c), the regular grid network becomes a random network with the least value of APL along with the reduced ACC value. Hence, in order to realize a small-world network, a limited number of LLs are sufficient as depicted in Figure 2.12(b).

Random Addition of New Links

In this link addition technique, a new LL in an already existing network is added between any two distant nodes based on the new link creation probability $p \in [0,\ 1]$ [4]. It can be noticed from the previous discussion that rewiring [3] involves removing one end of an existing NL and then connecting the open end to a long distance node in the network. However, rewiring is equivalent to dynamically changing the existing network topology, and thus, phase of the network is also changing continuously along with the changed direction of an NL. Random LL addition, on the contrary, does not involve removal of existing NLs. In Chapter 3, a detailed discussion on random link addition is carried out.

Euclidean Distance based Addition of New Links

An LL, in this link addition strategy, can be added based on the Euclidean distance or Manhattan distance [31, 32]. The Euclidean distance or Manhattan distance is measured between two points as the absolute difference in their coordinates. The probability of an LL addition is estimated by $p = d(u,\ v)^{-\alpha} / \sum_{v \neq u} d(u,\ v)^{-\alpha}$, where $d(u,\ v)$ is the Euclidean distance between node u and node v (see Figure 2.13), which is averaged over all node distances (the distant node pairs are at least $\geqslant 2$ hops apart) to get the normalized probability. Here α is the clustering exponent and takes the value equal to the network dimension. The observation revealed that for a 2-D grid network, $\alpha = 2$ gives the lowest value of the APL [32].

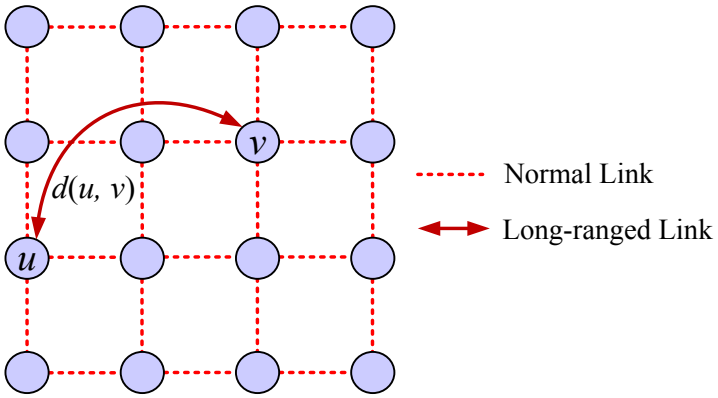


Figure 2.13: Addition of a few LLs are carried out based on the Euclidean distance in a 2-D lattice network. According to the LL addition strategy, an LL is added between nodes u and v based on the link addition probability $p = \frac{d(u,\ v)^{-\alpha}}{\sum_{u \neq v} d(u,\ v)^{-\alpha}}$, where α is the clustering exponent.

In the Euclidean distance based LL addition model [32], the probability of addition of LL was based on the Euclidean distance between the node pairs, and the clustering exponent α. However, the location information of the distant node is required to measure the Euclidean distance. Hence, some prior information about the network topology is required to implement the LL addition strategy.

2.3.4 Scale-Free Networks

A network is considered to be scale-free when the degree distribution of the network follows power-law. That is, the fraction of the nodes with degree D, that can be represented

as $P(D)$, is related as $P(D) \sim D^{-\gamma}$, where γ is the scaling exponent. An example power-law plot can be found in Figure 2.5(b). The relative nodal degree in a scale-free network greatly exceeds the average degree because of the existence of a few nodes with huge number of connections. The highly connected nodes are called *hub nodes*. Scale-free networks are very robust against random attacks, as there exists very low possibility to affect the hub nodes. Conversely, the scale-free networks are highly vulnerable to concentrated attacks because targeting a few hub nodes may turn a scale-free network dysfunctional.

An example scale-free network is shown in Figure 2.14, where most of the nodes have just a few links and a few nodes have large number of connections. The nodes with a large number of links, that is, the *hub nodes*, are depicted in the figure by a circle around the nodes.

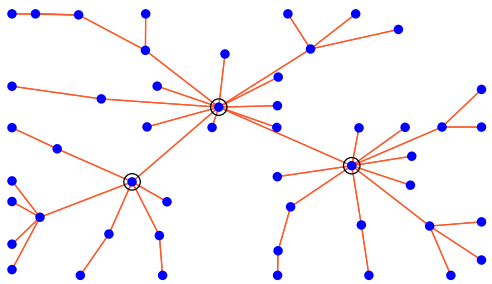


Figure 2.14: An example scale-free network of 50 nodes.

Network designers have explored many ways to create scale-free networks. The most common approaches are network formation (i) by preferential attachment, (ii) by a fitness based model, (iii) by varying intrinsic fitness, (iv) by local optimization of similarity and popularity, and (v) with exponent 1. However, we show in Chapter 3 that the greedy decision making, based on certain network metrics, such as APL, can also transform a regular network to a scale-free network.

Scale-Free Network Creation by Preferential Attachment

A network can evolve to a scale-free network when addition of nodes or links follows growth along with preferential attachment [5]. The preferential attachment strategy is also known as the Barabási-Albert (BA) model. In the BA network evolution model, while creating a few LLs, a newly introduced node in an existing network is more inclined to

connect to the nodes with higher degree values. Hence, high-degree nodes have higher chance to connect to other nodes in the network. A few nodes, according to the BA model, thus evolve as hub nodes over time.

Scale-Free Network Creation by Fitness based Modeling

In this form of scale-free network creation, fitness [33] plays a major role in network evolution. In general, all existing nodes in a network are not equally successful in acquiring new connections. In the preferential attachment based model, new nodes are mostly inclined to make connections with older nodes (those spend more time in the network). However, there are numerous examples where a newly introduced node gets many connections. For example, a new web page turns out to be very popular within a short timespan, a person becomes very influential within a community, or a research article receives a large number of citations in a very limited time. It can be seen from these networks that a few nodes turn out to be very popular or influential in a short timespan. The fitness based model can be studied to understand this kind of network evolution by incorporating a fitness parameter, to account for the sudden popularity, along with the BA model.

Scale-Free Network Creation by Varying Intrinsic Fitness

In the intrinsic nodal fitness model [34], a newly introduced node can be attached to an existing node based on certain characteristics of the node by which the attached node pair can mutually benefit. Note that the intrinsic fitness model does not take into account global information, such as degree based probability measure, which is the basic assumption in the preferential attachment and the intrinsic nodal fitness based network evolution models where certain fitness parameters can be considered along with the preferential attachment model. For instance, when a new person enters a community, he/she may not have information about individual connections. However, the person creates new links with other people based on certain behavioral characteristics or social influences by which the person thinks he/she may get some social advantages.

As it can be seen that the decision is solely based on the intrinsic characteristics of other nodes in a network, the network evolution model does not follow preferential attachment based link creation. This method also works differently from the fitness based

model, which considers fitness as one extra parameter in the growth and preferential attachment based network evolution.

Scale-Free Network Creation by Optimization

In this approach, a new node is added to the network based on the optimized score of the product of similarity and popularity of existing nodes in the network [9]. Here, popularity means how long a node has been in the network. Hence, popularity is similar to the fitness of a node in a network. Similarity is a measure of the closeness of a node with the newly introduced node in the network. That is, if a new node is closer to an existing node, the chance of connecting to the existing node is higher. The optimization framework takes care of the two above mentioned dynamics to gradually transforms a network to a scale-free network.

Scale-Free Network Creation with Exponent 1

In this scale-free network creation model, the scaling exponent (i.e., γ) of the network is evaluated to be 1 [35]. The realization of a scale-free network with exponent 1 is based on the equilibrium network models that produce the degree exponent value as 1. The network evolution model exercises the method of rewiring to transform a network to a scale-free network with exponent 1. In particular, the concept of random multiplicative process is applied to generate the scale-free model with $\gamma = 1$. In random multiplicative process, fluctuation of a random variable is proportional to its value in the system. In scale-free network modeling, the value of a random variable is taken as the degree of a node, and the fluctuation is realized by incorporating rewiring a connection at each time-slot, and the rewiring is proportional to the probability of the nodal degree in the network.

2.4 Summary

In this chapter, various complex network metrics, that are useful in exploring structure and dynamics of complex networks, are reviewed. The review was followed by a brief discussion of a few important complex network models such as regular networks, random networks, small-world networks, and scale-free networks. Hereafter, complex network metrics and network models are directly referred, wherever necessary, in rest of the thesis.

Chapter 3

Greedy Link Addition in Finite Sized Networks

Many real-world networks exhibit small-world characteristics where *small-world* refers to the fact that it requires only a small number of hops to reach a distant node from a source node. The main advantage of a small-world network is that it lies between a regular network and a random network, and incorporates beneficial characteristics from both the networks. The key characteristics of small-world networks include lower value of average path length (APL) and low to moderate value of average clustering coefficient (ACC). This chapter deals with the evolution of APL-optimal small-world networks, from regular networks, by introducing a few long-ranged links (LLs) with greedy decision making. Further, the gradual transition of a finite sized regular network to a fully connected network, with the addition of a few constrained LLs, is also studied toward the end of this chapter.

3.1 Existing Literature

In 1967, a social psychologist Stanley Milgram first observed the small-world (SW) characteristics in his message passing experiment [2] where it was found that any two acquaintances in the world can be connected by means of a very small number of intermediate acquaintances (median value being 5 to 9), popularly known as the *six degrees of separation*. In the late 1990s, the concept of SW characteristics became popular in several fields of study such as communication, transportation, and real-world social networks. Moreover, it was observed that many of the natural networks follow SW characteristics. Watts and Strogatz [3] first observed this feature in a few natural networks (e.g., neural networks of the bacteria Caenorhabditis Elegans, power-grid networks of the western USA, and the collaboration graph of film actors). SW characteristics of a network can be identified by low APL and low to moderate ACC. The concept of SW characteristics can also be used to design various networks.

In the following, a few existing literature on creation of small-world networks with (a) random decision and (b) deterministic approaches are discussed.

3.1.1 Creation of Random Small-World Networks

A small-world network can be realized by creation of random LLs between distant node pairs in a network. LLs can either be created based on rewiring of existing normal links (NLs) or by adding new LLs.

Rewiring of Existing Links

Rewiring is one of the techniques by which a regular network can be transformed to a small-world network. In rewiring, a few existing NLs are rewired to other nodes based on the rewiring probability p where $0 \leqslant p \leqslant 1$ [3, 36]. Here, a regular network (with probability p=0) is converted to a completely random network (p=1). In the transformation from a regular to a complete random network, small-world characteristics can be observed for $0 < p < 1$. In the process of rewiring a small number of existing NLs, lower value of APL can be attained. However, ACC is kept nearly unchanged compared to the regular network. A detailed discussion on rewiring technique can be found in Section 2.3.3.

Pure Random Addition of New Links

It can be seen that rewiring [3] involves removal of one end of an existing NL and then connecting the open end to a long distant node in the network. However, in pure random addition new LLs are added between two distant nodes in a network based on the link creation probability $p \in [0,\ 1]$ [4]. That is, two distant nodes are randomly selected with a certain probability q and a link is then added between the node pair if the link addition probability satisfies the condition $q \leqslant p$. The operational complexity of pure random link addition technique is lower compared to the rewiring of the existing NLs. Section 3.2 discusses LL addition based on the pure random decision making.

New Link Addition based on Euclidean Distance

In this LL addition strategy (also known as Kleinberg's model), a new link is added based on the Euclidean distance or Manhattan distance [31, 32]. The probability of an LL addition can be estimated as $p = d(u,\ v)^{-\alpha} / \sum_{v \neq u} d(u,\ v)^{-\alpha}$, where $d(u,\ v)$ is the Euclidean distance between nodes u and v. Here, α is the clustering exponent which represents the network dimension. That is, for example, $\alpha = 3$ when a 3-D network is concerned. Kleinberg's observation revealed that for a 2-D grid network, $\alpha = 2$ returns the lowest value of the APL. However, as location information of a distant node is essential to measure the Euclidean distance, some prior information about the network topology is required to implement the LL addition strategy.

The above discussed small-world network evolution strategies create new LLs based on the random decisions. In the following, a few existing deterministic LL addition strategies to realize small-world networks are briefly reviewed.

3.1.2 Creation of Deterministic Small-World Networks

Here we discuss a few deterministic LL addition strategies where LLs are added in a network to achieve certain performance benefits or to optimize certain network characteristics in the context of network performance.

Small-World Creation based on Constant and Variable Degree

This small-world model deterministically converts a regular network to a small-world network. The deterministic model [37] talks about two such approaches, constant and variable degree based small-world network creations. In the constant degree based small-world network creation model, a node in an N-node arbitrary network is replaced by a fully connected network. The fully connected network consists of same number of neighbor nodes (say k) the replaced node had. The resulting deterministic small-world network has $k \times N$ nodes along with the diameter of $2r + 1$ and the clustering coefficient of $\frac{N-2}{N}$, where $r \simeq \log_{k-1} N$. Conversely, in variable degree based small-world network creation the original arbitrary network is kept intact. However, fully connected networks

of variable sizes are attached to each node of the base network.

Small-World Creation based on Edge Iteration

The iterative new link creation strategy [38] to realize a deterministic small-world network is as follows: at time $t = 0$, there is a triangle shaped network. At time $t = 1$, each link of the triangle makes connection to a new node and hence, both the nodes of the edge get connected to the newly introduced node. Therefore, at $t = 1$, three new nodes are introduced and total six new links are added to the network. Similarly, at $t = 2$, six nodes are introduced and the connections are created as mentioned in the case of $t = 1$.

The deterministic small-world network, evolved in this iterative manner, contains a total of 3×2^k nodes and $3 \times 2^{k+1} - 3$ links after the iteration step at $t = k$.

Small-World Creation based on Various Optimized Network Parameters

To create a deterministic small-world network, an LL can also be added in a network to optimize certain network characteristics. However, the deterministic LL addition involves exhaustive search of possible locations of LLs and thus, time complexity for finding an optimal LL location is high [39]. The small-world networks, in this category, can be constructed based on various strategies such as minimize APL (MinAPL), minimize average edge length (MinAEL), maximize betweenness centrality (MaxBC), or maximize closeness centrality (MaxCC) metrics. The resultant small-world networks, with these LL-addition strategies, optimize when APL, AEL, BC, or CC performances are concerned.

In the following, random as well as greedy decision based LL addition strategies are discussed. Note that the simulation study is conducted on finite sized string topology networks (STNs) of varying sizes. The primary reason for choosing string topology, as the primary network to add LLs, is that there exists only limited literature that studied the properties of finite sized STNs. String topology is one of the most sparse network topologies and can be deployed to model practical networks such as highway communication networks, disaster response networks, and tactical networks. Moreover, STN can also be utilized to provide seamless wireless connectivity in rural or community networks where infrastructure networks are unavailable.

3.2 Random LL Addition

In this link addition technique, a new LL in an already existing network can be added between any two distant nodes based on new link creation probability $p \in [0,\ 1]$ or on the basis of pure random decision. In first method, a small number of new links are added with link addition probability p. That is, two distant nodes in a network are randomly selected with probability q and then an LL is created between the node pair, if the LL addition probability satisfies the condition $q \leqslant p$. The second method, on the other hand, randomly chooses two distant nodes and connects an LL if they are not connected already. Figure 3.1(a) depicts an example STN consisting of 40 nodes (only relevant nodes are shown). The random LL addition algorithm is listed in Algorithm 3.1.

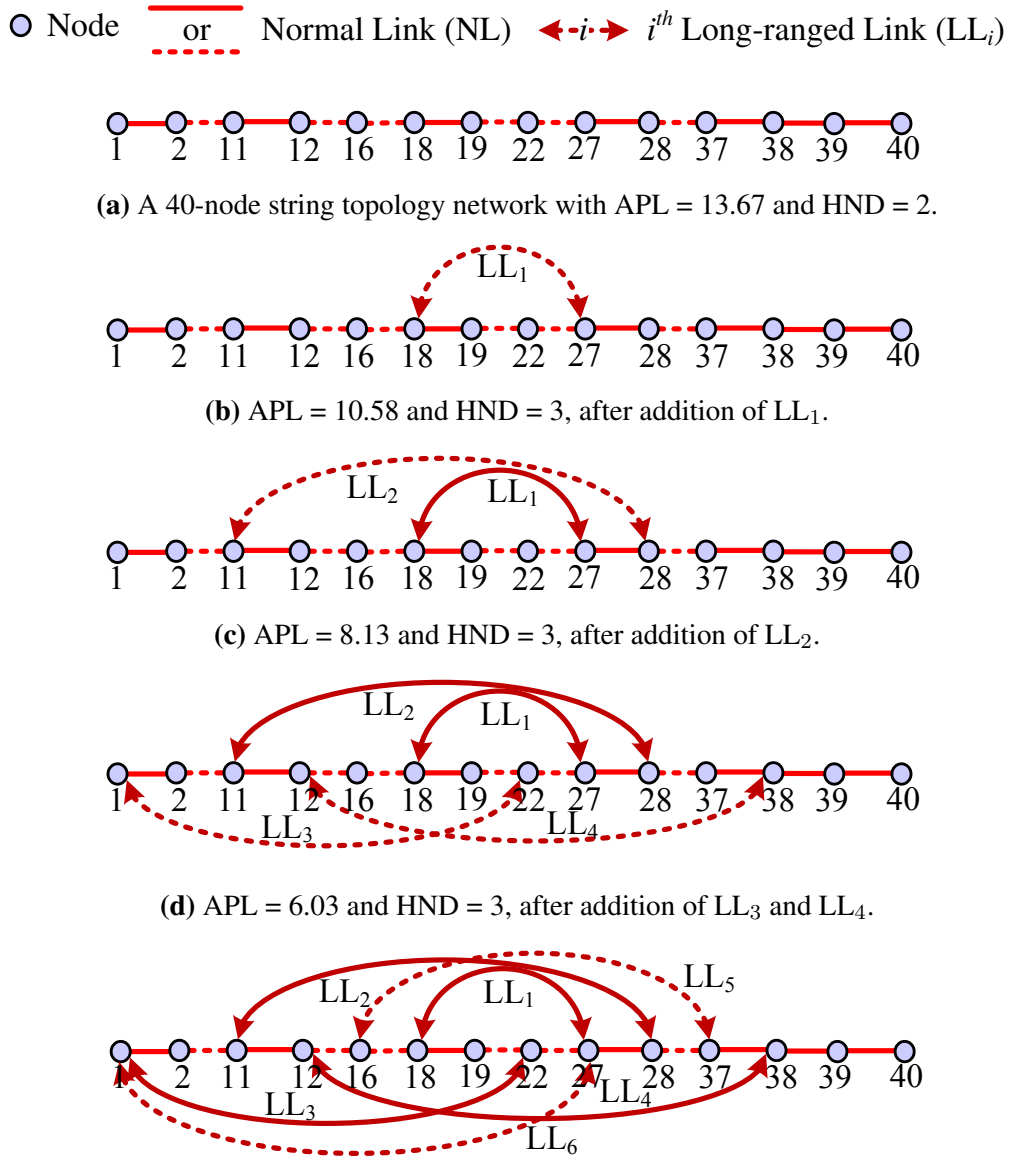
Algorithm 3.1 Random LL Addition

Require:

$\mathcal{G} = (\mathcal{V},\ \mathcal{E})$ — A network graph with $\mathcal{V}$ nodes and $\mathcal{E}$ edges
$(u,\ v)$ — A link between node u and node v
k — Number of LLs to be added in $\mathcal{G}$
$Possible_{LL}$ — A set of LL possibilities between node pairs in $\mathcal{G}$

1: **for** $i = 1 \rightarrow k$ **do**
2: $\quad Possible_{LL} = \{(u,\ v) \mid (u,\ v) \notin \mathcal{E}\}$
3: $\quad (u,\ v) \leftarrow random(Possible_{LL})$
4: $\quad \mathcal{E} \leftarrow \mathcal{E} \cup (u,\ v)$
5: **end for**

In Algorithm 3.1, a node pair in an N-node network is randomly chosen from all new LL addition possibilities and then an LL is added (lines 2-3). Algorithm 3.1 is $\mathcal{O}(k)$ time complex to add k LLs randomly in the network. Figures 3.1(b) through (e) show random addition of one, two, four, and six LLs in a 40-node STN. Figures 3.1(a) through (e) also depict the APL and the highest nodal degree (HND) values. Here, HND represents the degree of a node with the most number of connections in a network. It can be seen that the APL value decreases moderately as the LLs are added to the STN by incorporating the small-world characteristics.



(a) A 40-node string topology network with APL = 13.67 and HND = 2.

(b) APL = 10.58 and HND = 3, after addition of LL_1.

(c) APL = 8.13 and HND = 3, after addition of LL_2.

(d) APL = 6.03 and HND = 3, after addition of LL_3 and LL_4.

(e) APL = 5.16 and HND = 3, after addition of LL_5 and LL_6.

Figure 3.1: Pure random LL addition strategy is shown in an example undirected 40-node STN. (a) The STN before the pure random LL addition. The APL value is 13.67 along with the HND of 2. HND represents the node with the most number of connections. In (b), the first LL (LL_1) is added randomly between node 18 and node 27. The bi-directional dashed line represents newly connected LL. The APL value, after the addition of LL_1, is reduced to 10.58 and the HND changes to 3. Similarly, (c), (d), and (e) show the evolved network after the addition of two, four, and six LLs. It can be observed that the corresponding APL values are also reduced after the addition of new LLs.

3.2.1 Nodal Degree Distribution with Random LL Addition

The performance of the random LL addition strategy is studied in the context of various sized STNs (10 to 50 nodes). LLs are added in an N-node STN such that either the

network becomes fully connected or the STN accommodates $\frac{N}{2}$ LLs in the network. The nodal degree distribution is plotted, as can be seen in Figure 3.2, after the addition of moderate number of LLs with the random LL addition strategy.

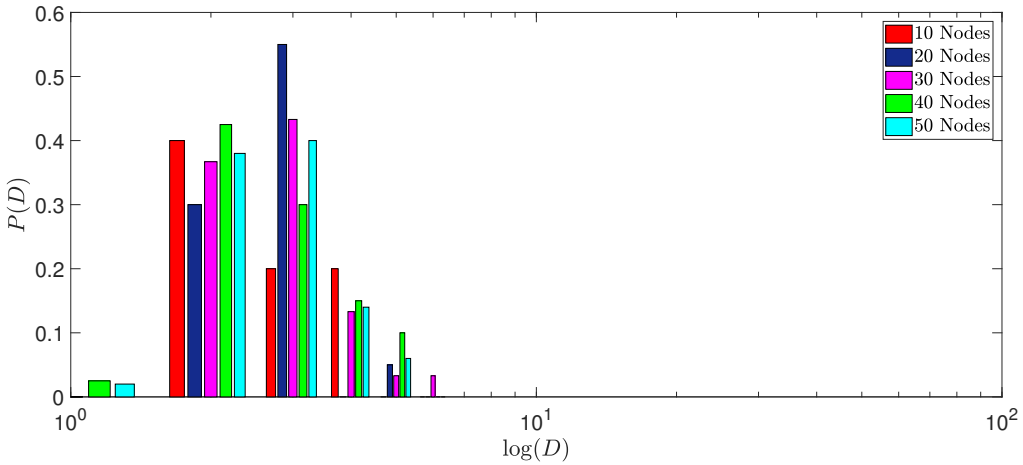


Figure 3.2: The degree distribution after the addition of $\frac{N}{2}$ LLs with the pure random strategy, with Algorithm 3.1, in the context of N-node STN. Plot of the degree distribution is in the semi-log scale where the 'X' axis represents degree value (D) of a node in logarithmic scale and the 'Y' axis depicts the probability of nodes with degree D ($P(D)$) in linear scale. The figure shows the degree distribution for 10, 20, 30, 40, and 50 node STNs where it can be noticed that, as most of the nodes have same degree, all distribution curves follow a distribution similar to the bell-shaped distribution.

In Figure 3.2, the degree distribution curves for five sets of results (10, 20, 30, 40, and 50 node STNs) are plotted. It can be seen from the figure that each degree distribution curve follows a bell-shaped pattern as the degree of most of the nodes are similar and centered around the mean of the distribution.

Table 3.1 lists the betweenness centrality (BC) values for all relevant nodes after the addition of each LL in Figure 3.1. The BC value identifies the most important node or a set of important nodes in a network. Higher the value of the BC of a node, more central the node is in a network. If an LL connects between a node pair with higher influences, the probability of formation of more number of shortest paths through the LL is also increased. As a result, the BC values of the node pair are also improved. In Table 3.1, BC values of a node pair is highlighted where an LL is added. That is, for example, first LL is added between nodes 18 and 27 and corresponding BC values are highlighted (shown in boldface) in the table. However, when LLs are added with pure random decision, it may not always find the best possible node pair locations to add an LL in order to maximize

Table 3.1
The BC values are shown for the relevant nodes in a 40-node STN, as shown in Figure 3.1(a), after the addition of each LL with the pure random decision as mentioned in Algorithm 3.1. Note that the BC value is improved for a node when it is connected to an LL (shown in boldface). However, as the LLs are added randomly in the network, significant improvement in the BC value of any particular node is not visible.

Results after LL addition with the pure random strategy

Total number of nodes = 40.

Number of LLs added = 6.

(Each LL connecting node pair is shown in boldface)

Nodes / **LL_i**	**1**	**2**	**8**	**11**	**12**	**14**	**16**	**18**	**20**	**22**	**26**	**27**	**28**	**32**	**37**	**38**	**39**	**40**
LL_1	0	0.05	0.30	0.39	0.42	0.46	0.49	**0.58**	0.09	0.02	0.13	**0.55**	0.44	0.33	0.15	0.10	0.05	0
LL_2	0	0.05	0.30	**0.48**	0.11	0.04	0.04	0.25	0.08	0.01	0.15	0.41	**0.64**	0.33	0.15	0.10	0.05	0
LL_3	**0.11**	0.08	0.15	0.33	0.10	0.03	0.05	0.26	0.12	**0.16**	0.14	0.38	0.55	0.33	0.15	0.10	0.05	0
LL_4	0.11	0.08	0.16	0.43	**0.33**	0.05	0.05	0.23	0.12	0.15	0.14	0.35	0.42	0.07	0.12	**0.24**	0.05	0
LL_5	0.11	0.08	0.15	0.34	0.21	0.02	**0.18**	0.26	0.14	0.16	0.12	0.26	0.31	0.05	**0.20**	0.18	0.05	0
LL_6	**0.26**	0.14	0.08	0.32	0.22	0.02	0.16	0.22	0.02	0.10	0.08	**0.43**	0.36	0.05	0.18	0.17	0.05	0

the BC value. It can be observed from the table that none of the nodes in the STN of Figure 3.1 has significantly higher value of the BC [23] when random decision based LL addition is used.

In random LL addition strategy, LLs are added in a network without any prior motivations such as optimizing the network performance or enhancing the network throughput. Random LL addition involves searching a node pair, based on certain probability, and creates an LL if the node pair is not already been connected. The LL addition strategy reduces the APL value of the STN, however, without guaranteeing the minimized value of the APL. The BC values of the nodes, as shown in Table 3.1, with respect to the STN with no LLs, are not significantly improved. Only the LL connecting nodes improve their respective BC values (shown in boldface in Table 3.1).

3.3 Greedy Decision based LL Addition

In order to achieve the objective of transforming a network to an APL-optimal small-world network, we propose greedy decision based LL addition strategy where an LL is added in a network such that the overall network APL can be minimized. The APL-optimal network is extremely efficient when minimizing the end-to-end transmission delay in a network is concerned.

The greedy decision based LL addition strategy can be subdivided into the following: (i) greedy optimal LL addition, and (ii) greedy near-optimal LL addition.

3.3.1 Greedy Optimal LL Addition

In the greedy optimal LL addition strategy,[1] an LL is created between a node pair in a network to deterministically optimize certain network characteristics. To add k LLs with the optimal LL addition strategy, all k LLs should be concurrently deployed in a network. Algorithm 3.2 lists the greedy optimal LL addition strategy to minimize the APL value of a network.

[1] In this chapter and rest of this thesis, the *greedy optimal* and the *optimal* are used interchangeably.

Algorithm 3.2 Greedy Optimal LL Addition

Require:
$\mathcal{G} = (\mathcal{V},\ \mathcal{E})$ — A network graph
k — Number of LLs to be added in $\mathcal{G}$

1: Check for all possibilities to get the lowest value of APL in $\mathcal{G}$ after adding k LLs
2: **if** More than one possibility exists to add k LLs **then**
3: Choose the only possibility which returns the minimum APL in $\mathcal{G}$
4: **end if**

Algorithm 3.2 searches for all possible LL locations, with brute-force, in order to add k LLs that minimize the APL value (line 1). If more than one possibilities exist to add the k LLs, then the optimal algorithm searches for each possibility and thus, after all searched combinations, the algorithm selects the best LL connecting locations (line 3).

Time Complexity of Greedy Optimal LL Addition Algorithm

The time complexity of the greedy optimal LL addition strategy (see Algorithm 3.2) can be determined as follows: To add an LL with the optimal strategy, N^2 possibilities exist in an N-node network. Further, in order to identify all-pair shortest paths (APSPs) in a network of N nodes, the optimal strategy uses Dijkstra's shortest path algorithm [40, 41] which is $\mathcal{O}(N^2 \log N)$ time complex to determine the APSPs. Therefore, the first optimal LL can be created with Algorithm 3.2 in $\mathcal{O}(N^2 \times N^2 \log N)$ or $\mathcal{O}(N^4 \log N)$ time. Likewise, to add two optimal LLs in an N-node network, there exist $\binom{N^2}{2}$ possibilities. Hence, the time complexity becomes $\mathcal{O}\left(\binom{N^2}{2} \times N^2 \log N\right)$ or $\mathcal{O}(N^6 \log N)$. Similarly, three optimal LLs can be created in $\mathcal{O}(N^8 \log N)$ time. To find the optimal locations for k LLs, the brute-force strategy has to be executed repeatedly. Therefore, to add k concurrent LLs optimally, Algorithm 3.2 takes $\mathcal{O}\left(\binom{N^2}{k} \times N^2 \log N\right)$ or $\mathcal{O}(N^{2k+2} \log N)$ time.

The implementation of Algorithm 3.2 is not a feasible solution in the context of real-time deployment of the APL-optimal LLs. In the following, an efficient greedy near-optimal deterministic LL addition strategy is proposed.

3.3.2 Greedy Near-Optimal LL Addition

The greedy near-optimal algorithm,[2] that is presented in Algorithm 3.3, is an efficient deterministic LL addition strategy. The near-optimal strategy searches for the possible node pairs in an N-node network to add the i^{th} LL, with the greedy decision making, which gives the lowest value of the APL. However, if multiple LL-location choices are identified for the i^{th} LL, one node pair is selected randomly, as presented in Algorithm 3.3. Since the i^{th} LL position is identified before the $(i+1)^{th}$ LL, the number of choices for the $(i+1)^{th}$ LL is not as exhaustive as in the case of the greedy optimal LL addition. Therefore, the near-optimal solution uses the brute-force strategy in a limited and sequential manner compared to the optimal LL addition algorithm (Algorithm 3.2).

Algorithm 3.3 Greedy Near-Optimal LL Addition

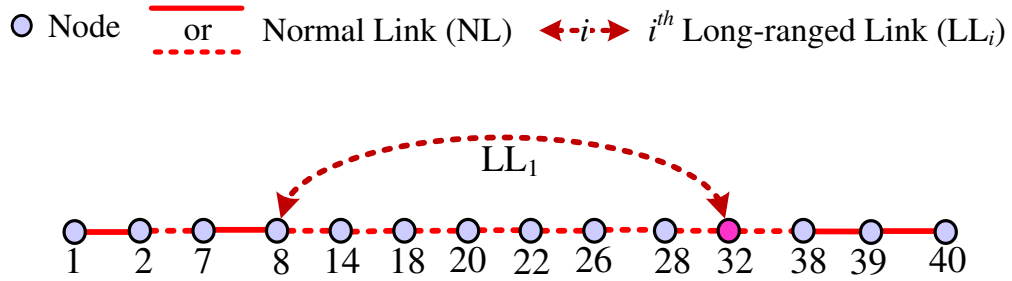
Require:
$\mathcal{G} = (\mathcal{V},\ \mathcal{E})$ — A network graph
k — Number of LLs to be added in $\mathcal{G}$

1: **for** $i = 1 \rightarrow k$ **do**
2: Check for every possibility to construct i^{th} LL over N nodes in $\mathcal{G}$
3: Estimate the APL of the network for each possibility at the last step
4: **if** More than one lowest APL LL possibility exists **then**
5: Randomly select one node pair having the least value of APL
6: **end if**
7: Add i^{th} LL between selected node pair giving lowest APL
8: Update network graph $\mathcal{G}$
9: **end for**

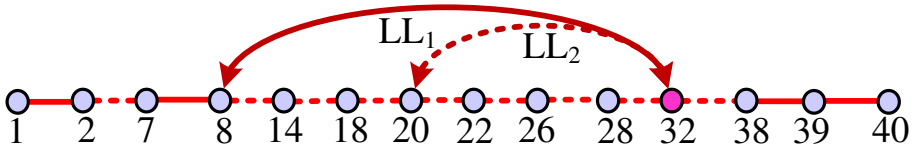
Algorithm 3.3 identifies LLs in a sequential manner by exhaustively searching for the N^2 LL connecting possibilities for a certain LL (lines 1-3). However, for multiple LL connection possibilities, the near-optimal strategy randomly selects an LL connecting location and thus, reduces the computational complexity (lines 4-6).

Figure 3.3 shows an example of greedy near-optimal strategy based LL addition where all the newly added LLs are shown in bidirectional dashed lines in an STN of 40 nodes. The greedy LL addition strategy finds node 8 and node 32 to connect the first LL (LL_1) in the network. The greedy LL addition strategy optimally selects the $0.2N^{th}$ and $0.8N^{th}$

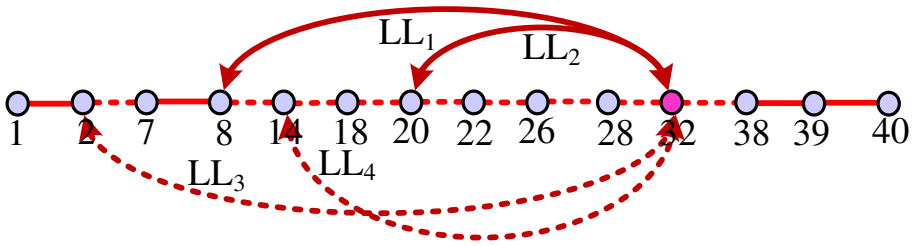
[2]In this chapter and rest of this thesis, the *greedy near-optimal* and the *near-optimal* are used interchangeably.



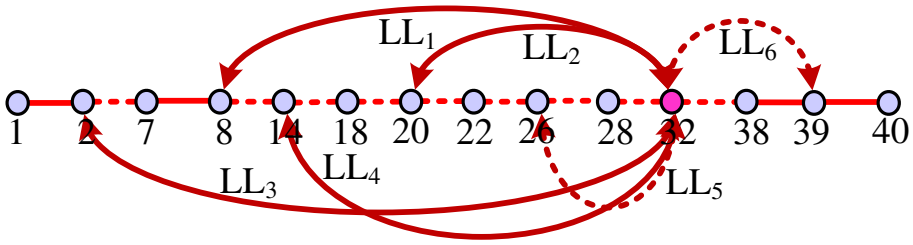
(a) APL = 8.41 and HND = 3, after addition of LL_1 in a 40-node string topology network.



(b) APL = 6.75 and HND = 4, after addition of LL_2.



(c) APL = 5.46 and HND = 6, after addition of LL_3 and LL_4.



(d) APL = 4.14 and HND = 8, after addition of LL_5 and LL_6.

Figure 3.3: An STN of 40 nodes where six LLs are added with the greedy near-optimal LL addition strategy. A new LL is shown in the dashed bidirectional line. The APL and the HND values are also mentioned in the figure. (a) The near-optimal LL addition strategy finds nodes 8 and 32, which are the $0.2N^{th}$ and $0.8N^{th}$ nodes looking from node 1 to N (in this example, $N = 40$), to add the first LL. (b) The second LL (LL_2) connects node 20 with node 32 which is the common node of LL_1 and LL_2. (c) The third and the fourth LLs (i.e., LL_3 and LL_4) select node 32 as one of the LL connecting nodes. (d) Similarly, the fifth and the sixth LLs also find node 32 as one of the LL end points, and thus node 32 gradually emerges as a hub node.

nodes looking from node 1 to connect LL_1. The location of LL_1 is found to be valid for the larger networks too. The nodes at $0.2N$ and $0.8N$ are called the *anchor nodes*. With the greedy near-optimal LL addition strategy, new LLs are always inclined toward one of the anchor nodes, where LL_1 is connected, and this phenomenon is termed as the *long-ranged link affinity* (LRA). The LRA continues in the network and one of the anchor nodes attracts more and more LLs and thus turns out to be a *hub node*.

In Figure 3.3(b), it can be seen that the near-optimal addition of the second LL (LL_2) also finds node 32 as one of the LL connecting nodes. Similarly, addition of four sequential LLs are shown in Figure 3.3(c). For each LL addition (i.e., addition of LL_3 and LL_4), the LRA influences the LL to select one of the anchor nodes with a substantially large number of connections in the network. As six or more LLs are added using the greedy near-optimal LL addition (Figure 3.3(d)), a hub node emerges at node 32 and thus the network gradually transforms to a scale-free network. It can be observed that as more number of LLs are added in a network with the near-optimal decision, a small-world network gradually transforms to a scale-free network by introducing hub node(s) in the network.

Time Complexity of Near-Optimal Algorithm

The time complexity of Algorithm 3.3 can be estimated as follows: As the APSP algorithm runs in $\mathcal{O}(N^2 \log N)$ time [41], the greedy near-optimal strategy requires $\mathcal{O}(N^2 \times N^2 \log N)$ or $\mathcal{O}(N^4 \log N)$ time to add an LL in an N-node network. Therefore, when k LLs are added sequentially with the near-optimal strategy, total computational complexity becomes $\mathcal{O}(k \times N^4 \log N)$ which is much lower than the greedy optimal LL addition strategy discussed in Algorithm 3.2.

3.3.3 Degree Distribution

The degree distribution, after the LL addition with the greedy near-optimal strategy, for 10, 20, 30, 40, and 50 nodes STNs are shown in Figure 3.4. The plots show the $P(D)$ values with respect to the $\log(D)$ values, where D denotes the number of neighbor connections of a node in a network, and $P(D)$ represents the probability of nodes with de-

gree D. It can be observed from the figure that the curves follow a power-law distribution (i.e., $P(D) \sim D^{-\gamma}$ [5]) with the scaling exponent value $\gamma \simeq 1.80$ [11].

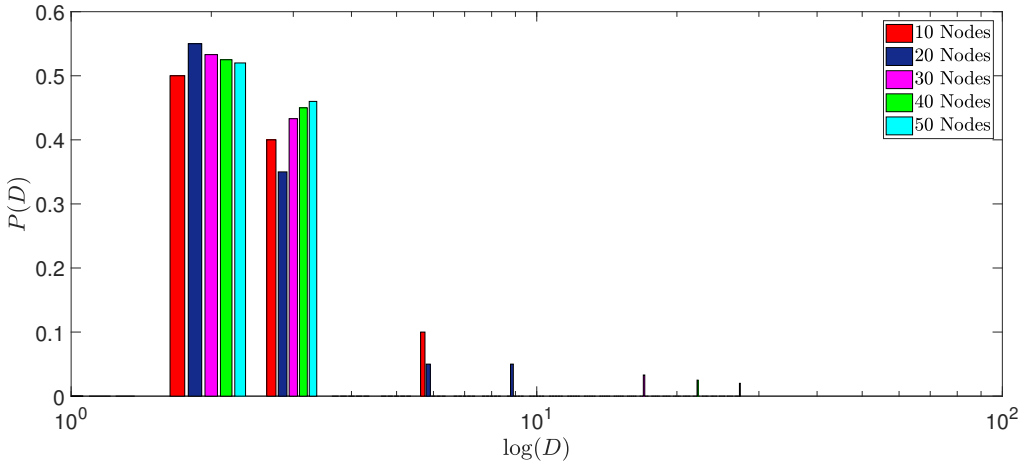


Figure 3.4: The degree distribution after the addition of $\frac{N}{2}$ LLs with the greedy near-optimal strategy in different sized STNs. The degree distribution is plotted in the semi-log scale. The figure shows degree distribution for 10, 20, 30, 40, and 50 node networks where it can be noticed that one of the anchor nodes is evolved as the hub node due to the LRA phenomenon and the degree distribution follows the power-law. Hence, the near-optimal LL addition transforms an STN to a scale-free network. The value of the scaling exponent for the power-law degree distribution is found to be $\gamma \approx 1.80$.

After the addition of each LL, one of the anchor nodes emerges as the node with the largest value of the betweenness centrality (BC) [23], and the node gradually exhibits *the rich get richer* principle by attracting more number of LLs. Table 3.2 shows the BC values of the network nodes mentioned in Figure 3.3. Note that BC values of LL connected nodes are highlighted (shown in boldface) after each LL addition in the network. From Table 3.2 it can be seen that, with the greedy near-optimal decision based LL addition, node 32 has evolved as the hub node as new LLs are more inclined to connect to the one of the anchor nodes (node 32 in a 40-node STN) with higher BC score.

It can be observed that the greedy decision making is one of the possible reasons behind the evolution of many real-world networks. For example, a small number of airports are directly connected to many other airports when the air traffic networks are concerned. The identification of those few airports, which are acted as the hub nodes, are mostly based on the greedy decision making of certain metrics such as the availability of the passengers, minimization of the travel time, and minimization of the fuel consumption. Hence, the evolution of a few hub nodes transforms the air-traffic network into a scale-free network.

Table 3.2
The BC values are shown for the relevant nodes of a 40-node STN after the addition of each LL with the greedy near-optimal LL addition strategy as depicted in Algorithm 3.3. Note that only the nodes with added LLs improve their BC values. The table lists the BC values of the nodes which are mentioned in Figure 3.3).

Results after LL addition with the greedy near-optimal strategy

Total number of nodes = 40.

Number of LLs added = 6.

(Each LL connecting node pair is shown in boldface)

Nodes / **LL_i**	**1**	**2**	**8**	**11**	**12**	**14**	**16**	**18**	**20**	**22**	**26**	**27**	**28**	**32**	**37**	**38**	**39**	**40**
LL_1	0	0.05	**0.51**	0.26	0.24	0.20	0.16	0.12	0.09	0.12	0.20	0.22	0.24	**0.53**	0.15	0.10	0.05	0
LL_2	0	0.05	0.46	0.11	0.08	0.03	0.07	0.15	**0.37**	0.14	0.02	0.05	0.09	**0.66**	0.15	0.10	0.05	0
LL_3	0	**0.15**	0.30	0.11	0.07	0.03	0.08	0.15	0.38	0.14	0.02	0.05	0.09	**0.70**	0.15	0.10	0.05	0
LL_4	0	0.16	0.22	0.01	0.03	**0.21**	0.03	0.03	0.30	0.13	0.02	0.05	0.09	**0.76**	0.15	0.10	0.05	0
LL_5	0	0.16	0.22	0.01	0.03	0.21	0.03	0.03	0.21	0.03	**0.20**	0.05	0.01	**0.79**	0.15	0.10	0.05	0
LL_6	0	0.16	0.22	0.01	0.03	0.21	0.03	0.03	0.21	0.03	0.20	0.03	0.01	**0.80**	0.03	0.07	**0.16**	0

Further, the business supply chain management networks, the author citation networks, and the telephone exchange networks are also some of the real-world examples where the networks are evolved to scale-free networks by means of the greedy decision making [1].

Therefore, it can be seen from the previous discussion that the APL-optimal small-world STNs (SWSTNs) may not be achieved with the random decision based LL addition strategy. To achieve the APL-optimal SWSTNs, the greedy decision based LL addition strategy can be exercised.

3.4 Length Constrained LL Addition

It can be observed from the last section that a few LLs can be added with the greedy near-optimal decision, in a finite sized network, to achieve an APL-optimal small-world network. However, we imposed no restriction to those cases on the length of an LL. In this section, we study the evolution of a finite sized network when a few LLs are added with certain imposed constraints while achieving the APL-optimal small-world network. The term *constrained LL* means that the creation of an LL obeys certain rules at the time of deployment in a network. In our study, the LLs are created with the restriction on the maximum length. That is, for instance, with the maximum length of five hops, no constrained LL with more than five hops is allowed to be added in a network. Note that the minimum length of an LL is always assumed to be 2 hops.

We study the evolution of finite sized STNs as well as grid topology networks when a set of LLs are added with the greedy near-optimal decision to minimize the network APL. The performance of the near-optimal decision based constrained LL addition is measured in STNs of varying sizes (ranging from 50 to 200 nodes) and grid topology networks (ranging from 10×10 to 25×25 nodes). We add $\frac{N}{2}$ and $\frac{N^2}{2}$ constrained LLs in the context of an N-node STN and $N \times N$ node grid network. In case of an STN, the length of a constrained LL (LL_{MaxLen}) varies from 2 hops to $\frac{N}{2}$ hops. Conversely, $\mathrm{LL}_{MaxLen2D}$ ranges from 2 hops to $\frac{N}{2}$ hops when an $N \times N$ grid network is concerned.

3.4.1 Constrained LL Addition in Finite String Topology Networks

In order to study various network characteristics, $\frac{N}{2}$ constrained LLs are added with the greedy near-optimal strategy in an N-node STN, where $N \in \{50,\ 100,\ 150,\ 200\}$. In the following, we study the nodal degree and the the average length of an LL after the addition of $N/2$ constrained LLs in various sized STNs.

Observations on Nodal Degree

Figure 3.5 plots the maximum degree (D_{max}), the second maximum degree (D_{max2}), and the third maximum degree (D_{max3}) values in the context of STNs of varying sizes. The observation is carried out to identify the gradual evolution of an STN with respect to the varying LL_{MaxLen} values.

It can be observed from the figure that the differences between the D_{max} value and the D_{max2} as well as the D_{max3} values are significant when the LL_{MaxLen} value reaches toward $\frac{N}{2}$. As the LL_{MaxLen} values are increasing, a certain node gradually evolves as the hub node by attracting most of the constrained LLs in the network. The evolution results in a scale-free network. From Section 3.3, it can also be observed that as the LL_{MaxLen} value reaches $0.6N$, the first optimal LL in an N-node STN is always connected between the anchor nodes situated at the $0.2N^{th}$ and $0.8N^{th}$ fractional locations.

However, when $0.3N < LL_{MaxLen} < 0.4N$, the D_{max} and D_{max2} values have minimal deviation while the D_{max3} value greatly deviates from the rest. The observations also indicate the presence of multiple hub nodes and the network behaves as a scale-free network. Conversely, when $LL_{MaxLen} \leqslant 0.2N$, the network acts similar to a small-world network (as no hub node emerges) where the differences among the D_{max}, D_{max2}, and D_{max3} values are minimal.

Average Length of a Constrained LL

Figure 3.6 shows plots of the average length of an LL (D_{avg}) after the addition of all LLs in an STN with different LL_{MaxLen} values. It can be seen from the figure that the average length of an LL is approximately $0.6 \times LL_{MaxLen}$. Moreover, there is a minimal change in

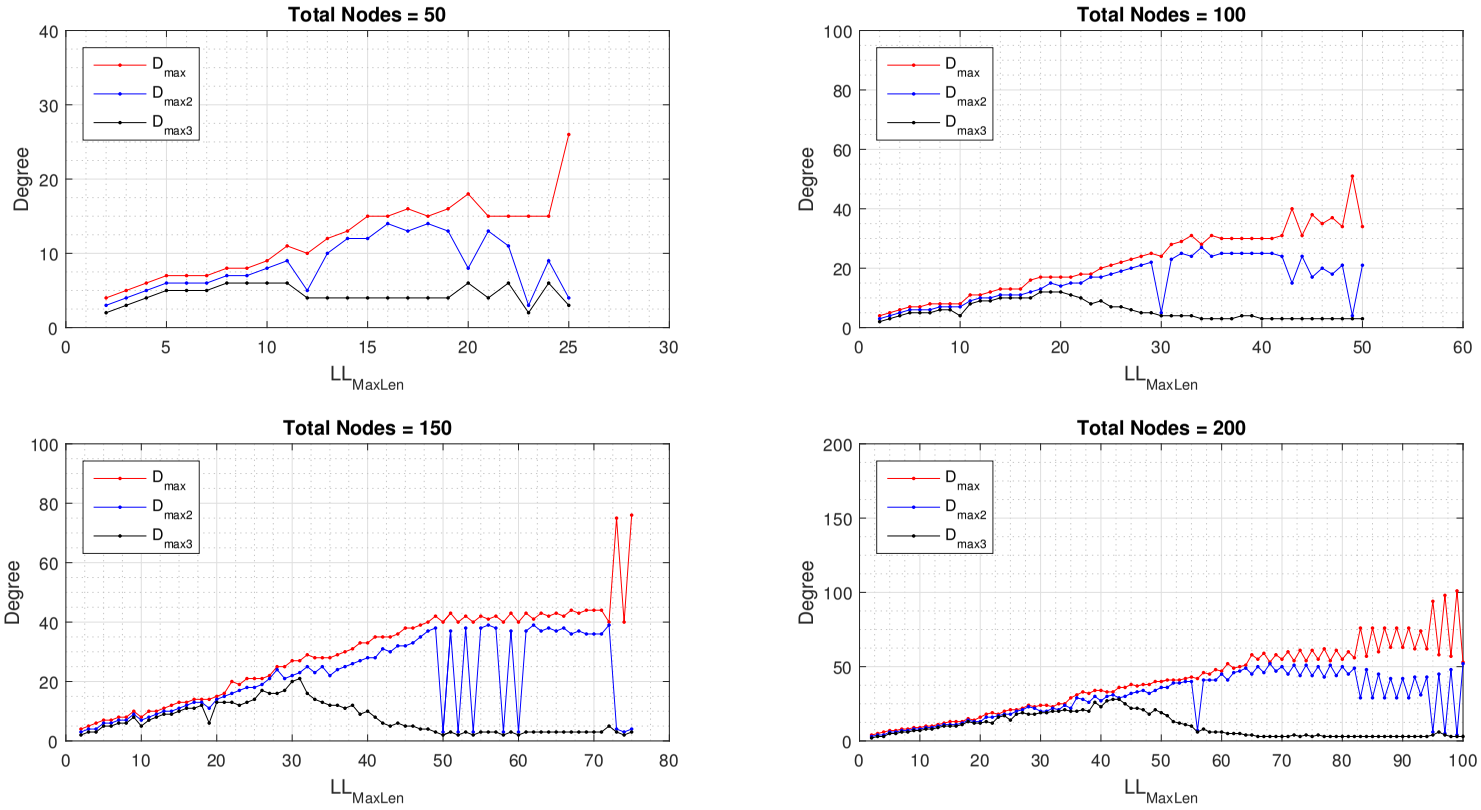


Figure 3.5: Plot of the maximum degree (D_{max}), the second maximum degree (D_{max2}), and the third maximum degree (D_{max3}) with respect to different LL_{MaxLen} values for 50, 100, 150, and 200 node STNs.

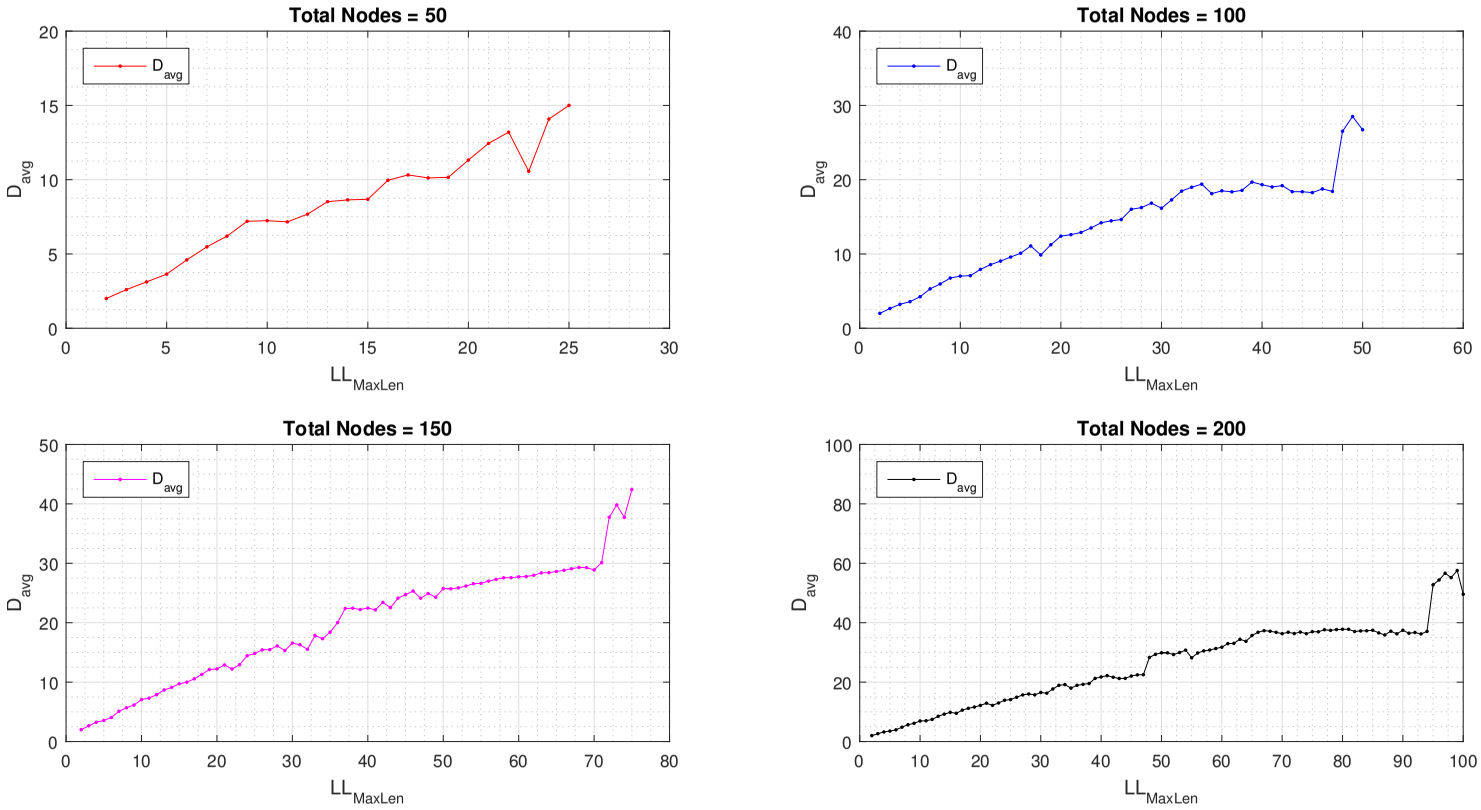


Figure 3.6: Plot of the average length of an LL (D_{avg}) with respect to different LL_{MaxLen} values for 50, 100, 150, and 200 node STNs.

the D_{avg} value in the range $0.3N < LL_{MaxLen} < 0.4N$. This observation is valid because the STN behaves as a small-world network when $0.3N < LL_{MaxLen} < 0.4N$.

3.4.2 Constrained LL Addition in Finite Grid Topology Networks

We also add $\frac{N^2}{2}$ constrained LLs with the greedy near-optimal strategy in an $N \times N$ grid network where $N \in \{10,\ 15,\ 20,\ 25\}$. The maximum value of an LL in a grid network is in the range $LL_{MaxLen2D} \in [2,\ \frac{N}{2})$. Note that, in case of a grid network, the maximum value of the $LL_{MaxLen2D}$ value is equal to the row-size of the grid network.

Observations on Nodal Degree

Figure 3.7 shows plots of the maximum nodal degree (D_{max}), the second maximum nodal degree (D_{max2}), and the third maximum nodal degree (D_{max3}) after the addition of $\frac{N^2}{2}$ constrained LLs.

It can be seen from the figure that the D_{max} values show significant differences with respect to the D_{max2} values when the $LL_{MaxLen2D}$ value approaches $\frac{N^2}{2}$. Therefore, a certain node in a grid network gradually attracts most of the LLs and evolves as a hub node. However, when the $LL_{MaxLen2D}$ value ranges between $0.3N$ and $0.6N$, multiple hub nodes can be evolved in the network and thus, the regular grid is gradually transformed to a scale-free network. Moreover, with the $LL_{MaxLen2D}$ values in the range between 2 and $0.3N$, the grid network behaves as a small-world network. Note that, in the simulation, the maximum value of the $LL_{MaxLen2D}$ is kept as N which is the row-size in a grid network. This particular setting of the maximum value of the $LL_{MaxLen2D}$ is due to the fact that once the $LL_{MaxLen2D}$ value becomes greater than the row-size in a grid network, a certain node attracts most of the LLs and gradually transformed to a hub node.

Average Length of a Constrained LL

Figure 3.8 shows the plot of the average length of a certain LL after the addition of all LLs in grid topology networks of varying sizes with different $LL_{MaxLen2D}$ values. It can be observed that the average length of an LL is approximately $0.8 \times LL_{MaxLen2D}$. Further,

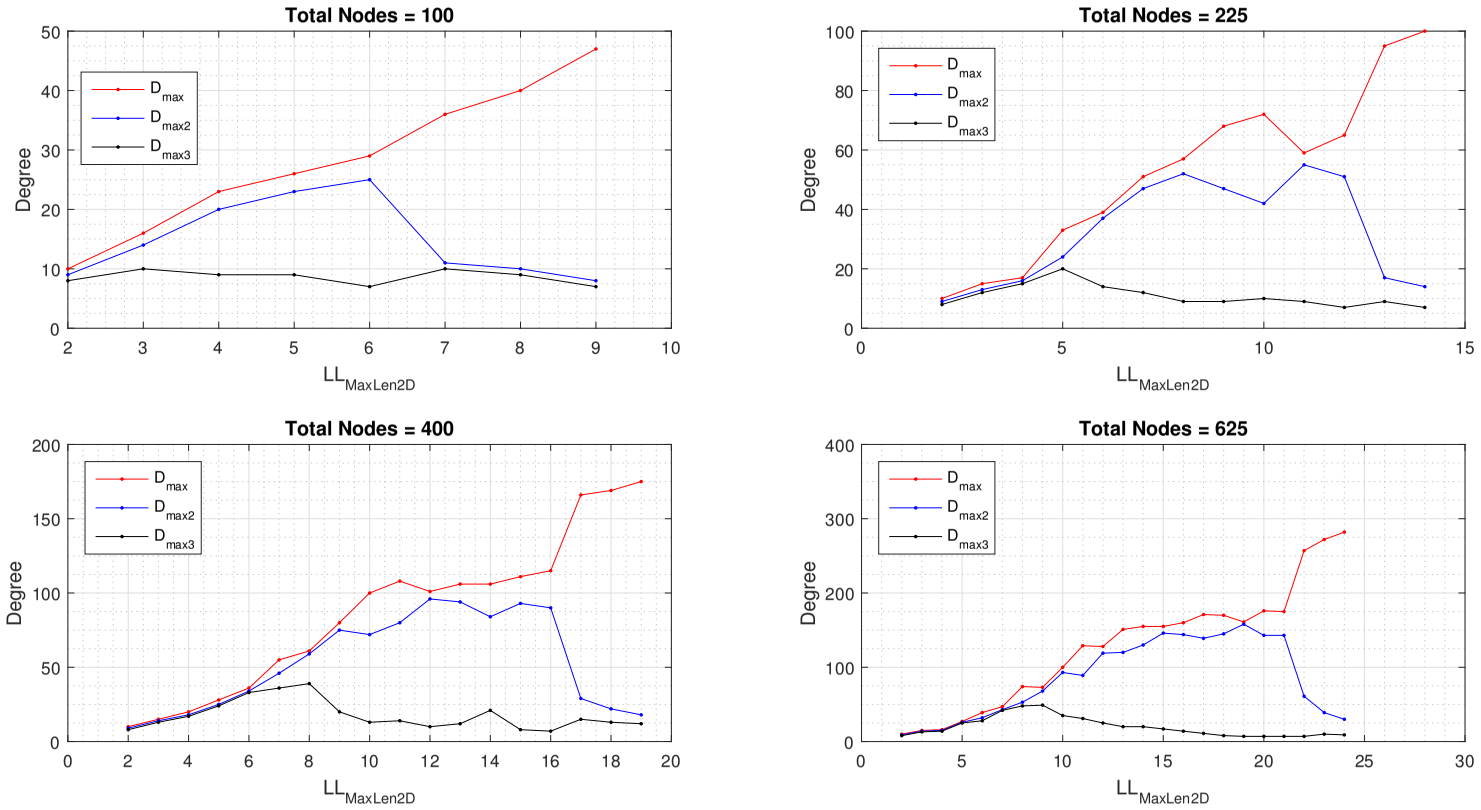


Figure 3.7: Plot of the maximum degree (D_{max}), the second maximum degree (D_{max2}), and the third maximum degree (D_{max3}) with respect to different $LL_{MaxLen2D}$ values for 10×10, 15×15, 20×20, and 25×25 node grid networks.

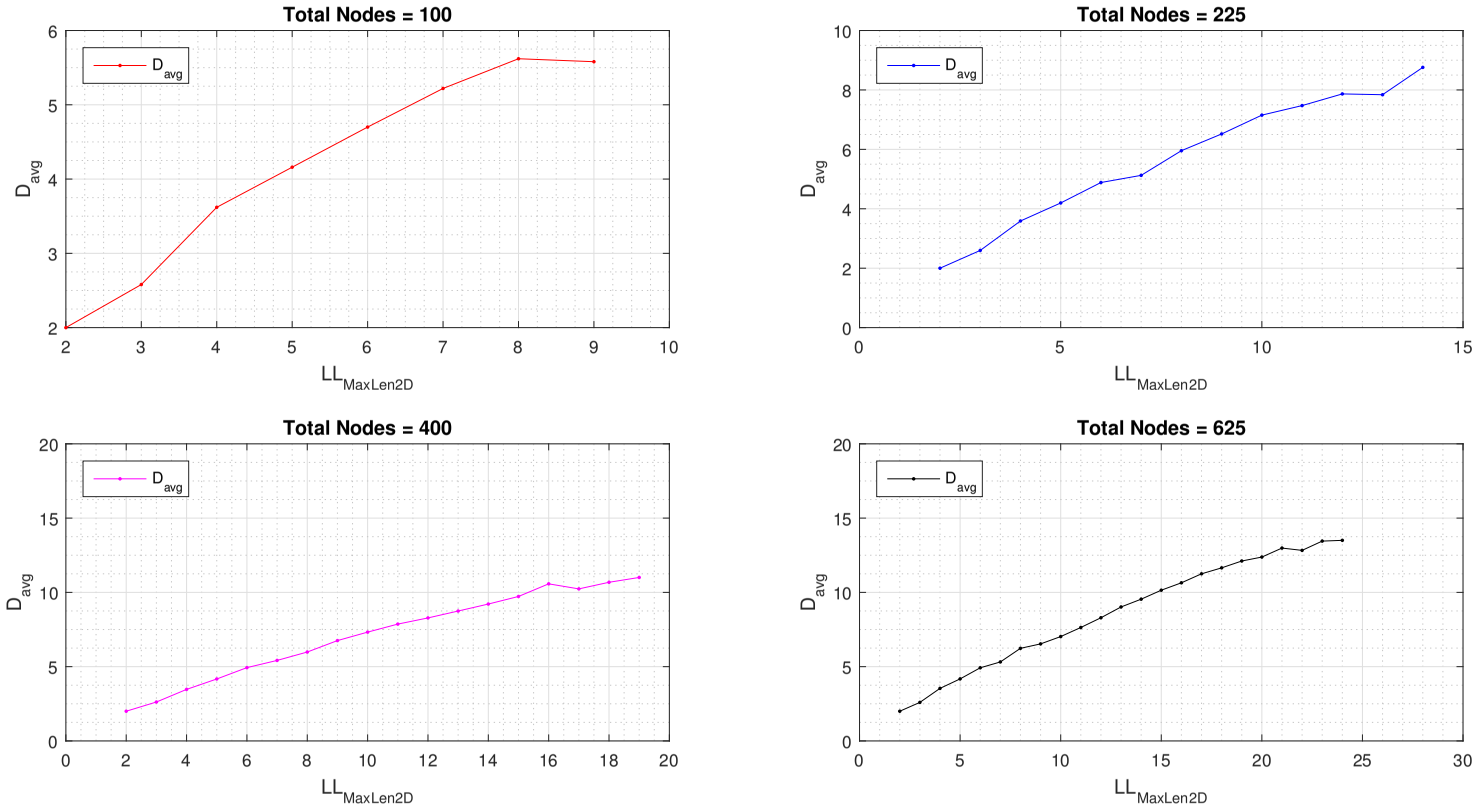


Figure 3.8: Plot of the average length of an LL (D_{avg}) with respect to different $LL_{MaxLen2D}$ values for 10×10, 15×15, 20×20, and 25×25 node grid networks.

there is a minimal change in the D_{avg} value as the $LL_{MaxLen2D} \rightarrow N$. This observation can also be validated from the plot of the maximum nodal degree, in Figure 3.7, where it can be seen that as the $LL_{MaxLen2D}$ value approaches the size of the row of a grid network, a single hub node is gradually emerged.

From the above discussion on greedy near-optimal constrained LL addition in finite sized string as well as grid networks to optimize the network APL, it can be observed that there is a gradual transition in the network as follows: A finite sized regular network such as a string or a grid network is gradually transformed to a small-world network with the optimal-APL. As more LLs are introduced to the network, a few nodes in the network thus turn out to be the hub nodes by attracting most of the LLs and, the network is gradually transformed to a scale-free network with the potentially truncated degree distribution. The degree distribution truncates because the network size (i.e., the total number of nodes) is fixed, and with time, as the number of the LLs increases, the scale-free behavior gradually fades away and the network shifts toward a fully connected mesh. It can be seen that the evolution of a finite sized network to a fully connected network takes place in the following manner: A regular network $\rightarrow$ a small-world network $\rightarrow$ a scale-free network with the truncated degree distribution $\rightarrow$ a fully connected network.

3.5 Summary

In this chapter we studied the evolution of the APL-optimal small-world networks by adding a few LLs with random as well as the greedy near-optimal decision. It was observed that during the transformation of a finite sized STN of N nodes to an SWSTN, the first LL always finds a unique location between the anchor nodes situated at the $0.2N^{th}$ and $0.8N^{th}$ fractional locations. However, the random decision based LL addition could not achieve the APL-optimal SWSTN. Moreover, we further studied the evolution of the finite sized STNs as well as the grid topology networks by adding the length constrained APL-optimal LLs. We observed from the simulation results that a finite sized network evolves in the following way: A regular network $\rightarrow$ a small-world network $\rightarrow$ a scale-free network with the truncated degree distribution $\rightarrow$ a fully connected network.

Chapter 4

Analytical Identification of Anchor Nodes

In the last chapter, while creating a few long-ranged links (LLs) with the greedy near-optimal decision to achieve average path length (APL) optimal small-world string topology network (STN), it was observed that the first LL always finds an optimal location between the $0.2N^{th}$ and $0.8N^{th}$ node pair in an N-node STN. We termed the fixed fractional locations of the nodes as the *anchor nodes* because one of the nodes attracts most of the LLs, when added with the greedy decision making, and gradually evolves as the hub node. In this chapter we develop an analytical model to identify the fixed fractional locations of the anchor nodes in a string topology network. We also validate our observations on the importance of the anchor nodes through exhaustive simulation study.

4.1 Existing Literature

The problem of adding new LLs to a network can be formulated as an edge addition problem for a graph [31, 42, 43, 44], where the graph models the connectivity of a network. A graph $\mathcal{G}$ consists of a set of N nodes and $\mathcal{E}$ links[1] such that a node corresponds to a network entity, and a link represents connectivity between two entities in the network. LL addition in order to minimize the APL has been considered in [31, 42, 43, 44].

In [31], a decentralized link addition algorithm was proposed under the assumption that the edge-length between two neighboring nodes is related to the Euclidean distance between the nodes. The authors of [42] obtained the edge addition algorithms which approximately minimize the weighted network APL. The authors also observed that for the problem of minimizing the total path length for a single source, the optimally added links will all be incident on the source node. The authors proposed an approximation strategy for APL minimization for a general graph, where a source node is chosen which

[1] We use *links* and *edges* in this chapter, and rest of this thesis, interchangeably.

is close to all other source nodes, and links are added in order to minimize the total path length for the chosen node. In [43], constant factor approximation algorithm was developed to identify possible edges to be added in order to minimize the network APL. In [44], a path screening strategy was also proposed to obtain possible edges to be added.

We proposed our greedy near-optimal algorithm [11] to minimize the APL for an N-node STN (see Chapter 3). It has been found that the time complexity to add an APL-optimal LL with the greedy near-optimal decision is $\mathcal{O}(N^4 \log\ N)$ (see Section 3.3.2). It is important that at least for certain simple graphs, which find applications in modeling a wide variety of real-world networks, some approximations are obtained. These approximations are significant since they potentially lead to good initial points for search and optimization algorithms, which can be used for further exact optimal LL addition. The string (linear) graph is used to model practical networks such as highway communication networks, disaster response networks, rural community networks, and tactical networks. For an STN, we already observed that the network after the greedy near-optimal LL addition had a simple structure where the positions of hub nodes as a fraction of N is approximately fixed even N is varied. Since the hub nodes anchor the optimally added links, their identification in the initial graph is important. Figure 4.1 shows a 40-node STN where six LLs are added to achieve APL-optimal small-world network. It can be seen from the figure that one of the nodes, where the first optimal LL is added, gradually attracts other optimal LLs and evolves as a hub node.

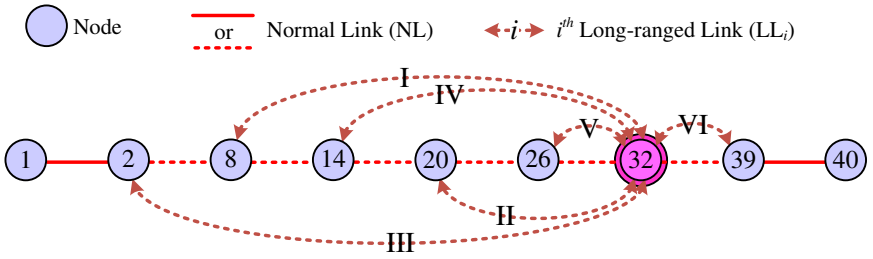


Figure 4.1: Addition of six LLs (I to VI) to a 40-node STN to minimize the APL. It has been seen that the optimal LL addition leads to a single hub node in the STN (see node 32). In this figure, the hub node is shown by a *dark colored* filled double circle.

4.2 Significance of Anchor Nodes

The identification of anchor nodes in an STN can serve many purposes. For example, in a community broadband network with string topology, one of the anchor nodes can serve as a gateway to the Internet and thus, overall data transmission time in the community broadband network can be improved. When an ad hoc communication network is deployed as part of a military communication or during an emergency response situation, APL-optimal network can be created by adding a few LLs in the network with one of the anchor nodes as hub node.

The identification of anchor nodes is also beneficial when vehicle-to-vehicle communication in platoons [45] is concerned. In a platoon,[2] the platoon controller must be knowledgeable about the real-time positioning, vehicle speed, and overall fuel consumption of other vehicles to efficiently control the transportation network. This real-time information is transmitted to the controller over unreliable wireless channel, leading to significant delay. By identifying the anchor nodes and, thereby, creating an LL with the directional antenna systems, the delay can be minimized to enhance the controller accuracy for maintaining the platooning systems.

In social networks, where the linear ordering of nodes in a string graph could model nodes ordered according to their addition times to the network or a hierarchy among nodes, anchor nodes can be considered as epochs in time or places in the hierarchy to which other nodes should attach in order to enhance the network influence. Hence, anchor nodes play a crucial role in communication and social networks.

4.3 Identification of Anchor Nodes

In Chapter 3, while sequentially adding LLs with the greedy near-optimal decision, it was observed that the added LLs were all incident on a single node in the network, which behaves similar to a hub node. Moreover, it was observed that the fractional position of

[2] A platoon, which is a smart technology based transportation system, consists of a number of vehicles equipped with sensor devices with which one vehicle can closely follow and communicate with other vehicles.

the hub node, that is, the node index as a fraction of N, is constant for $N \geqslant 10$. In Figure 4.1, we have shown an example of a 40-node STN where six edges are added with near-optimal strategy and we observe that node 8, that is, the $0.2N^{th}$ node, becomes a hub node (shown by a *filled* double circle). It was found that the first LL always connects between the $0.2N^{th}$ and the $0.8N^{th}$ nodes in an N-node network. We refer to the node at this fixed fractional position as an *anchor node*.

We note that the fractional positions of the anchor nodes could be obtained by finding out the nodes on which the first optimally added link is incident. In the following, we analytically identify the fixed fractional positions of the nodes by considering APL optimal addition of the first link for a dense string network obtained in the asymptote of large N.

4.3.1 Problem Statement

We assume that $\mathcal{G} = (\mathbf{V},\ \mathbf{E})$ is a string graph where $\mathbf{V}$ is set of nodes, with $|\mathbf{V}| = N$, and $\mathbf{E}$ is set of edges. Let nodes p_1 and p_2 be such that if an LL is added between p_1 and p_2, the graph APL is minimized. From the earlier discussion, we have that either p_1 or p_2 would be the anchor node. Therefore, in order to identify the anchor nodes obtained by APL optimal addition of LLs to a string graph, it is sufficient to find out where a single LL should be added (to identify p_1 and p_2) in order to minimize the graph APL.

We analytically identify the nodes p_1 and p_2 by considering a dense string graph. A dense string graph is of unit diameter where infinitely large numbers of nodes are present. Formally, a dense string graph can be obtained as a limit of a sequence of string graphs, each with nodes $(1, 2, \ldots, N)$, as $N \to \infty$. It can be assumed in a dense string graph that the distance between the nodes i and $i + 1$, as well as the LL to be added between nodes p_1 and p_2, is $\frac{1}{N}$. We note that as far as optimization of APL is concerned, scaling all distances in a graph by the same quantity does not matter. The scaling which we have chosen leads to a limit graph consists of a continuum of nodes in the interval [0, 1] (see Figure 4.2(a)).

The single edge addition problem for this limit graph can be formulated as follows. We assume that the LL is added between the positions p_1 and p_2 for the limit graph where $p_1,\ p_2 \in (0,\ 1)$ and $p_1 < p_2$. For the limit graph in Figure 4.2(a), suppose the shortest

path length between nodes s and d is $p\,(s,\ d)$. We note that $p(s,\ d)$ is a function of p_1 and p_2. By a change of variables, $p(s,\ d)$ can be written as $p(s,\ x)$, where $x = d - s$. Then the total path length of all shortest paths starting from s (and considering only those d to the right of s) is $\int_{x=0}^{1-s} p(s,\ x)dx$. Then the total path length is

$$\int_{s=0}^{1} \int_{x=0}^{1-s} p(s,\ x)dxds. \tag{4.1}$$

We note that minimization of the APL for a given graph is equivalent to minimizing the total path length. Hence, our problem is to find p_1^* and p_2^*, optimal points for p_1 and p_2, respectively, that minimize the total path length in Equation (4.1).

4.3.2 Anchor Nodes for a Dense String Network

We first obtain APL for the limit graph in Figure 4.2(a) as an explicit function of p_1 and p_2. Then we show that the minimization of APL is a convex optimization problem for which unique optima p_1^* and p_2^* can be obtained. We analytically identify p_1^* and p_2^*, and find that they are approximately the same as that observed in the greedy near-optimal LL addition [11].

The function $p(s,\ x)$ is different depending on whether $s \in [0,\ p_1]\,,\ (p_1,\ p_2)\,,\ [p_2,\ 1]$. Therefore, we consider Equation (4.1) as the sum of the three integrals, S_1, S_2, and S_3, where each term corresponds to the integral of s in $[0,\ p_1]$, $(p_1,\ p_2)$, or $[p_2,\ 1]$. Each case is discussed in the following.

Case 1:

When $s \in [0,\ p_1]$, the following subcases arise (see Figure 4.2(a)):

1. When $s + x \leqslant p_1$, $p\,(s,\ x) = x$.
2. When $p_1 < s + x \leqslant p_2$, then

$$\begin{aligned} p\,(s,\ x) &= \min\,(x,\ p_1 - s + p_2 - (s + x)) \\ &= \min\,(x,\ p_1 + p_2 - 2s - x)\,. \end{aligned}$$

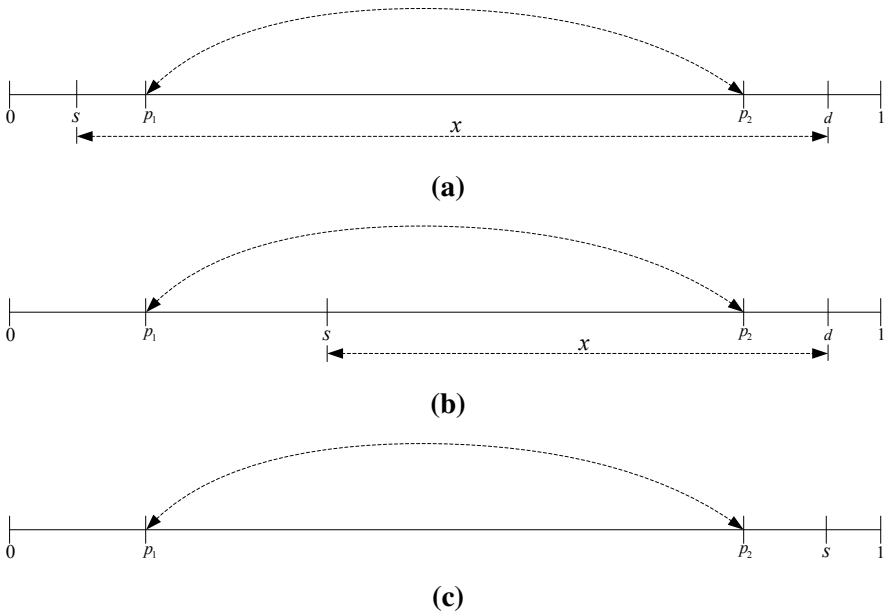


Figure 4.2: A dense string graph obtained as the limit of a string graph with N nodes as $N \rightarrow \infty$ is shown in (a). APL is optimized with respect to the fractional positions p_1 and p_2 of the two anchor nodes where the first optimally added LL is connected. The shortest path length $p(s,\ d)$ is different depending upon whether s is in $[0,\ p_1]$, $(p_1,\ p_2)$, $[p_2,\ 1]$. These three cases are shown in (a), (b), and (c). In (a), s is placed at the left side of p_1 (i.e., Case 1), in (b), s is in between anchor node p_1 and anchor node p_2 (i.e., Case 2), and in (c), s is at the right side of p_2 (i.e., Case 3).

3. When $p_2 < s + x$

$$\begin{aligned} p\,(s,\ x) &= p_1 - s + (s + x) - p_2 \\ &= x + p_1 - p_2. \end{aligned}$$

Then we have that

$$S_1 = \int_0^{p_1} \int_0^{1-s} p(s,\ x) dx ds, \tag{4.2}$$

where $\int_0^{1-s} p(s,\ x)dx$ is

$$\int_0^{p_1-s} x dx + \int_{p_1-s}^{\frac{p_1+p_2}{2}-s} x dx + \int_{\frac{p_1+p_2}{2}-s}^{p_2-s} (p_1 + p_2 - 2s - x)\, dx + \int_{p_2-s}^{1-s} (x + p_1 - p_2)\, dx.$$

Case 2:

When $s \in (p_1,\ p_2)$, there are two possibilities (see Fig. 4.2(b)) for the shortest path $p\,(s,\ x)$:

1. When $s + x \leqslant p_2$,

$$\begin{aligned} p\,(s,\ x) &= \min\,(x,\ s - p_1 + p_2 - (s + x)) \\ &= \min\,(x,\ p_2 - p_1 - x)\,. \end{aligned}$$

2. When $s + x > p_2$,

$$\begin{aligned} p\,(s,\ x) &= \min\,(x,\ s - p_1 + (s + x) - p_2) \\ &= \min\,(x,\ 2s + x - p_1 - p_2)\,. \end{aligned}$$

Similarly, using Equation (4.1) we get S_2 as

$$\int_{p_1}^{p_2}\int_{0}^{1-s} p\,(s,\ x)\,dxds = \int_{p_1}^{\frac{p_1+p_2}{2}} Y ds + \int_{\frac{p_1+p_2}{2}}^{p_2}\int_{0}^{p_2-s} xdxds + \int_{\frac{p_1+p_2}{2}}^{p_2}\int_{p_2-s}^{1-s} xdxds, \tag{4.3}$$

where $Y = \int_0^{\frac{p_2-p_1}{2}} xdx + \int_{\frac{p_2-p_1}{2}}^{p_2-s} (p_2 - p_1 - x)\,dx$.

Case 3:

When $s \in [p_2,\ 1]$ (see Figure 4.2(c)) we get from Equation (4.1) that

$$S_3 = \int_{p_2}^{1}\int_{0}^{1-s} xdxds. \tag{4.4}$$

Combining Equation (4.2), Equation (4.3), and Equation (4.4), we obtain that

$$\begin{aligned}\int_{s=0}^{1}\int_{x=0}^{1-s} p(s,\ x)dxds &= S_1 + S_2 + S_3 \\ P(f_1,\ f_2) &= \frac{5}{24}\left(p_2^3 - p_1^3\right) - \frac{1}{4}\left(p_2^2 - 3p_1^2\right) + \\ &\quad \frac{3}{8}p_1p_2\left(p_2 - p_1 - \frac{4}{3}\right) + \frac{1}{6}. \end{aligned} \tag{4.5}$$

Therefore, our problem is to

$$\begin{aligned} &\text{minimize} && P(p_1,\ p_2) \\ &\text{such that} && p_1,\ p_2 \in (0,\ 1), \\ & && p_1 < p_2. \end{aligned} \tag{4.6}$$

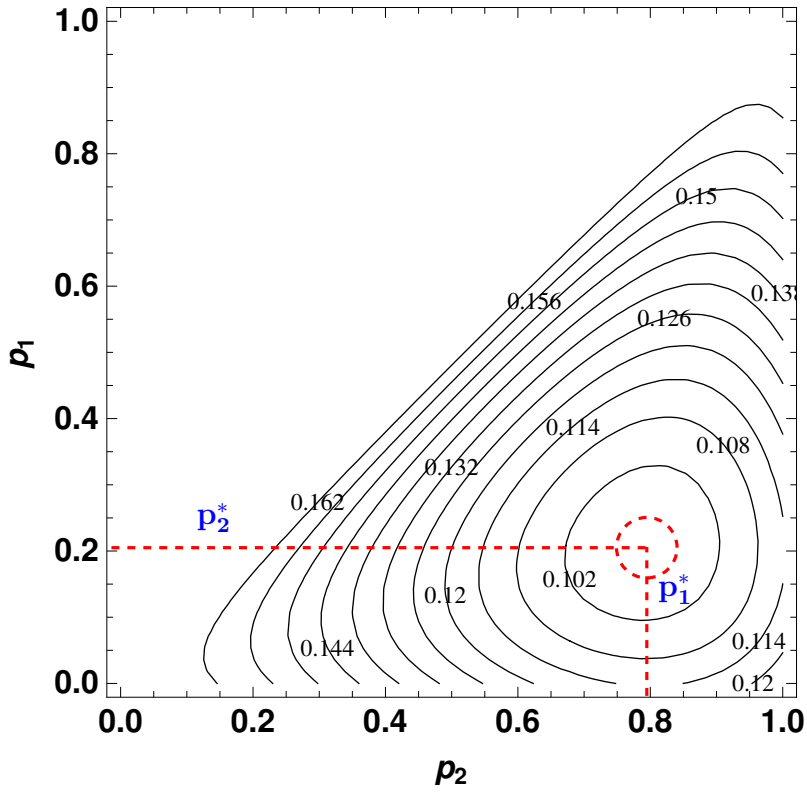


Figure 4.3: Objective function $P(p_1,\ p_2)$ in the constraint set $p_1 < p_2$.

We check whether $P(p_1,\ p_2)$ is convex by considering the determinant of the Hessian (H) of Equation (4.5)

$$det(H) = \left| \begin{bmatrix} \frac{3}{2} - \frac{5}{4}p_1 - \frac{3}{4}p_2 & \frac{3}{4}p_2 - \frac{3}{4}p_1 - \frac{1}{2} \\ \frac{3}{4}p_2 - \frac{3}{4}p_1 - \frac{1}{2} & \frac{5}{4}p_2 + \frac{3}{4}p_1 - \frac{1}{2} \end{bmatrix} \right| \geqslant 0.$$

We note that $P(p_1,\ p_2)$ is a strictly convex function (since H is positive definite) in the constraint set and can be solved in constant time.

The optimal values of p_1 and p_2, that is p_1^* and p_2^*, respectively, can be obtained from Equation (4.6), as 0.2071 and 0.7929 which are unique [46]. Figure 4.3 shows 2-D contour plot of the objective function $P(p_1,\ p_2)$. From the analytical derivation, therefore, it can be seen that the fractional positions of the anchor nodes for a dense string topology network coincides with the observation of the greedy near-optimal strategy based APL-optimal LL addition [11].

4.4 On the Locations of Anchor Nodes

In order to understand why the fixed fractional position of the anchor node is either at $0.2N$ or $0.8N$, we compare APL-optimal LL addition of an STN with that for a ring topology network (see Figure 4.4(a)). We assume that the distance between neighboring nodes for the ring network and that for the STN network is 1.

An APL-optimal LL can be added between the $0.25N^{th}$ and $0.75N^{th}$ nodes, as can be found with the greedy near-optimal strategy, for the ring network. Now suppose the edge between 1 and N for the ring network is removed to obtain a string network (see Figure 4.4(b)), then the nodes at which the APL-optimal LL is added is shifted from $0.25N$ to $0.2N$ and $0.75N$ to $0.8N$ (see Figure 4.4(b)).

Table 4.1 shows the total path length (PL) experienced from different fractions of string as shown in Figure 4.5. It can be seen from Table 4.1 that $\text{PL}_{1-0.2N}$ (i.e., substring from node 1 to node $0.2N$) and $\text{PL}_{0.8N-N}$ are contributing equal fractions of the path length (24.8% with respect to the total network path length), whereas $\text{PL}_{0.2N-0.8N}$

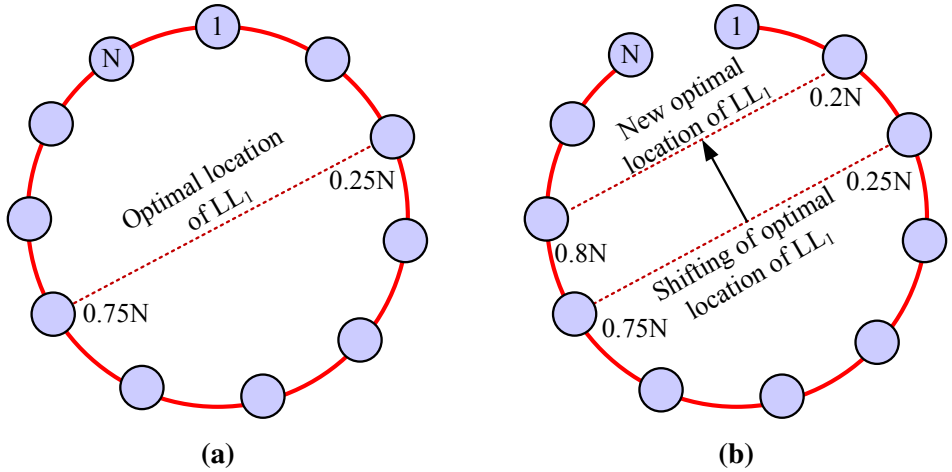


Figure 4.4: (a) An example ring topology of N nodes. (b) The ring topology can be converted to an STN by removing a link.

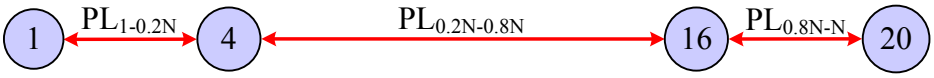


Figure 4.5: An example 20-node finite string topology. The fractions of $\text{PL}_{1-0.2N}$, $\text{PL}_{0.2N-0.8N}$, and $\text{PL}_{0.8N-N}$ of an N-node network are shown in the figure where PL is the total path length. Note that, for instance, $\text{PL}_{1-0.2N}$ is pointing to the substring from node 1 to node $0.2N$ (in this figure, node 4).

Table 4.1
Total path length experienced by different parts of string topology networks

Sl. No.	Number of Nodes	$\text{PL}_{1-0.2\text{N}}$	$\text{PL}_{0.2\text{N}-0.8\text{N}}$	$\text{PL}_{0.8\text{N}-\text{N}}$
1	20	660	1,340	660
2	40	5,288	10,744	5,288
3	60	17,852	36,276	17,852
4	80	42,320	86,000	42,320
5	100	82,660	167,980	82,660

fraction is contributing approximately 50.4% path length to the total path length of the network. Since the nodes toward the extremities of the STN contribute more to the total path length (and hence, the APL for a fixed N), the positions of the first optimally added LL are closer to the extremities for the case of the STN as compared to the ring network.

4.5 Influence of the Anchor Nodes

In the last section, we analytically determined the fixed fractional locations of the anchor nodes in an STN. We also mentioned that how the locations of the anchor nodes help in minimizing the total path length of an STN. To further study the importance of the anchor nodes, we perform simulation study in 30- and 40-node STNs where six LLs are added with the following strategies: (i) With random decision (i.e., Algorithm 3.1), (ii) with greedy near-optimal decision (i.e., Algorithm 3.3), and (iii) with the combination of the both random and greedy near-optimal strategies. We experiment with only six LLs because of the high time complexity of the greedy near-optimal decision based LL addition, as discussed in Chapter 3.

4.5.1 Based on Random LL Addition

When six LLs are added based on the random decision, it can be observed that no hub node is evolved in the network. Figure 4.6(a) shows a 40-node STN where six LLs are added with the random decision. It can be noticed from the figure that as LLs are added to the STN randomly, no hub node can be seen in the network. Table 4.2 also shows that the APL deviation is approximately 48.65% from the greedy near-optimal LL addition (where the hub node is at 0.8^{th}). As all LLs are added randomly to the network, the random LL addition strategy could not find any anchor node in the network. Therefore, random LL addition fails to transform the network to a scale-free network by attracting subsequent LLs to the anchor node and eventually transforms the anchor node to a hub node.

4.5.2 Based on Greedy Near-Optimal LL Addition

When six LLs are added with the greedy near-optimal decision, the first LL is always connected between the anchor nodes located at the 0.2^{th} or 0.8^{th} fractional locations in an STN. Figures 4.6(b) and 4.6(c) show the two situations where 0.2^{th} and 0.8^{th} nodes are found out to be the hub nodes, in a 40-node STN, to achieve the APL-optimal scale-free networks. The hub nodes are depicted in the figures as the *filled circle* nodes.

It can also be seen that the 0.8^{th} node becomes the automatic choice for the hub node (see Figure 4.6(c)). This is due to the higher betweenness centrality value [23]

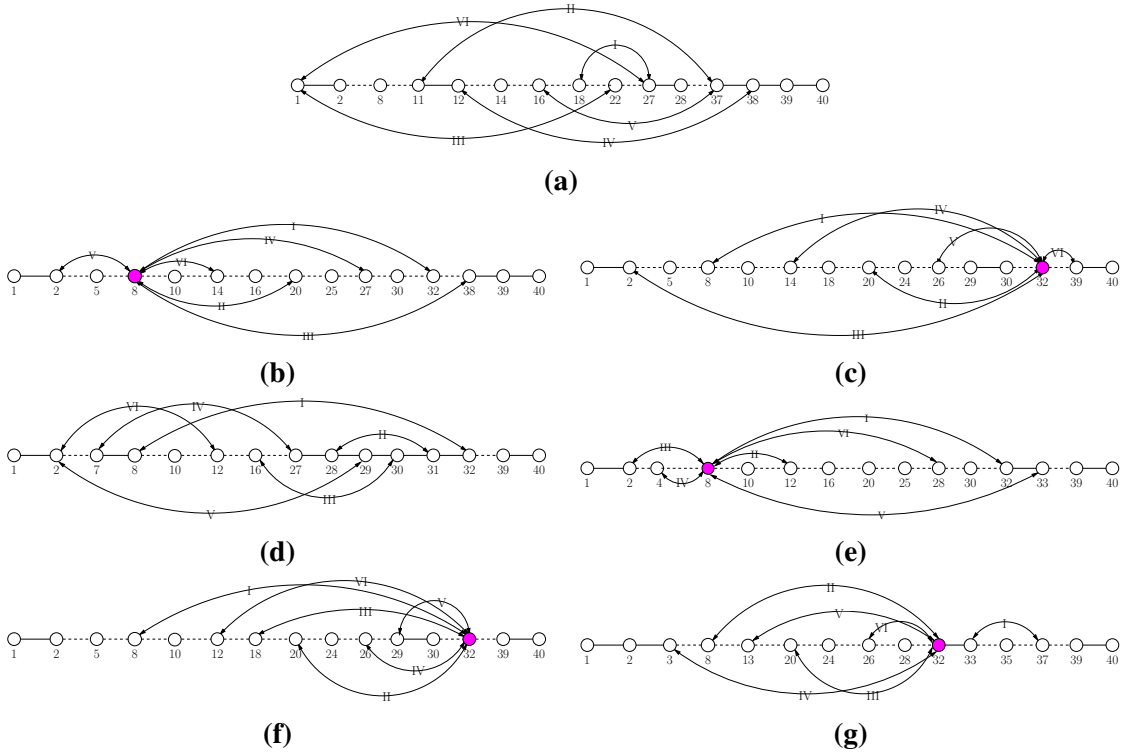


Figure 4.6: Six LLs are added, with different strategies, in an example 40-node STN. The evolved hub node in the STN is depicted as the *filled circle*. All corresponding APL values, after the addition of six LLs, are depicted in Table 4.2. (a) Six LLs are added with the random decision and we see that no hub node is created. (b) Addition of six LLs with the greedy near-optimal decision where node 8, that is, the 0.2^{th} node, evolves as the hub node. (c) Addition of six LLs with the greedy near-optimal decision where node 32, that is, the 0.8^{th} node, evolves as the hub node. (d) LL_1 is added based on the near-optimal decision which is followed by the random addition of five LLs. It can be observed that no hub node is formed as the subsequent LLs addition after the first optimal LL does not follow the objective of APL-optimal STN. (e) LL_1 is added based on the near-optimal decision and then subsequent LLs are added randomly by keeping 0.2^{th} node, that is node 8, as one of the LL-connecting nodes. (f) The strategy of (e) is followed by keeping 0.8^{th} node, that is node 32, as one of the LL connecting nodes. (g) LL_1 is added randomly and then subsequent LLs are added with the greedy near-optimal decision.

of the 0.8^{th} node than that of the 0.2^{th} node after the addition of LL_1. However, the APL deviation is very low, as can be seen from Table 4.2, if 0.2^{th} node is selected as the hub node (approximately, 0.35% for the 30-node network and 0.86% for the 40-node network).

4.5.3 Based on Combination of Both LL Addition

We study LL addition with different combinations of random and greedy near-optimal decisions in 30- and 40-node STNs. Figures 4.6(d) through (g) show addition of six LLs, in a 40-node STN, with the following strategies: (i) The first LL is added based on the greedy near-optimal strategy and the rest of LLs are added with random decision (see Figure 4.6(d)), (ii) the first LL is added with greedy near-optimal decision and other five LLs are added randomly by keeping the 0.2^{th} node as one of the LL-connecting nodes (see Figure 4.6(e)), (iii) the first LL is added with the greedy near-optimal decision and rest of the LLs are added randomly by keeping the 0.8^{th} node as one of the LL-connecting nodes (see Figure 4.6(f)), and (iv) the first LL is added randomly and rest of the LLs are added with the greedy near-optimal decision (see Figure 4.6(g)).

Table 4.2
APL values for different positions of anchor nodes in string topology networks

Total number of LLs added = 6							
No. of Nodes	**Random Addition (20 Seed)**	**Near-optimal Addition with hub node at** $0.2N^{th}$	**Near-optimal Addition with hub node at** $0.8N^{th}$	**Optimal LL$_1$ Remaining LLs with Random Addition**	**Optimal LL$_1$ Remaining LLs with Random Addition keeping hub node at** $0.2N^{th}$	**Optimal** LL_1 **Remaining LLs with Random Addition keeping hub node at** $0.8N^{th}$	**Random LL$_1$ Remaining LLs with Near-optimal Addition**
30	5.09	**3.39**	**3.41**	4.56	4.36	3.86	3.58
40	6.05	**4.10**	**4.07**	5.92	5.84	5.37	4.31

In Figure 4.6(d), LL$_1$ is added with the greedy near-optimal decision, and the remaining LLs are added randomly. It can be seen that no hub node is evolved in the 40-node STN due to the fact that no optimality criterion was involved while adding the random decision based LLs. Hence, anchor nodes only can attract LLs that are added with the objectives of APL-optimal transformation in the network.

Conversely, in Figures 4.6(e) and 4.6(f), 0.2^{th} and 0.8^{th} nodes are selected as one of the random LL-connecting nodes in the STN. In spite of selecting one of the anchor nodes as the hub node, it can be seen from Table 4.2 that the APL-optimal STN is not achieved because of the absence of greedy decision making in creating new LLs.

In Figure 4.6(g), the first LL is added randomly and the remaining five LLs are added with the greedy near-optimal decision. It can be observed that the 0.8^{th} node is evolved as the hub node in the network. However, the APL deviation from the LL addition strategy

of Figure 4.6(c) is approximately 5.93% in the 40-node STN.

From the above discussion it can be seen that to realize APL-optimal STNs, two criteria have to be considered: (i) Identification of the anchor nodes, and (ii) LL addition based on the greedy near-optimal decision making [47]. If one of the two criteria is not present, an STN cannot be transformed to an APL-optimal small-world STN. Further, near-optimal LL addition is also responsible for transforming one of the anchor nodes as the hub node where most of the APL-optimal LLs are getting connected.

4.6 Summary

The locations of the anchor nodes were identified when we added LLs with the greedy near-optimal decision in order to achieve APL-optimal small-world STNs. In this chapter, we analytically determined the fixed fractional locations of the anchor nodes, at 0.2071 and 0.7929, in an STN. We also found, through exhaustive simulations, that identification of anchor nodes as well as the greedy near-optimal decision based LL addition are responsible for realizing APL-optimal small-world STNs by transforming one of the anchor nodes to a hub node.

Chapter 5

Sequential Deterministic Long-Ranged Link Addition

Transforming a string topology network (STN) to an average path length (APL) optimal small-world STN is highly time complex. The greedy near-optimal decision based long-ranged link (LL) addition is not a feasible solution for faster network deployment scenarios, such as creation of ad hoc networks for military applications and setting up communication networks for highways or during natural calamities. In this chapter, a heuristic strategy called sequential deterministic long-ranged link addition (SDLA) is presented. SDLA efficiently creates a set of LLs in an STN to optimize the end-to-end hop distances between node pairs. This chapter discusses SDLA algorithm, its time complexity analysis, and performance comparison from the greedy near-optimal LL addition strategy.

5.1 Observations from Previous Study

From previous chapters we observed that the greedy near-optimal decision based LL addition transforms a finite sized STN to an APL optimal small-world STN (SWSTN). We also noticed that the first LL is always connected to the fixed fractional locations, known as the anchor nodes. However, the greedy near-optimal LL addition is $\mathcal{O}(k \times N^4 \log\ N)$ time complex to add k APL-optimal LLs in an N-node STN. Addition of six LLs in a 30-node STN with the near-optimal decision making (see Algorithm 3.3) is shown in Figure 5.1.

In Figure 5.1, it can be seen that the first LL always finds its location between the $0.2N$ and $0.8N$ nodes, also known as the anchor nodes, in an N-node STN. A detailed discussion about the anchor nodes can be found in Chapter 4. It can also be seen that the $0.8N$ node evolves as a hub node where rest of the LLs are connected to achieve an APL-optimal SWSTN. However, for a large value of N, the greedy near-optimal strategy

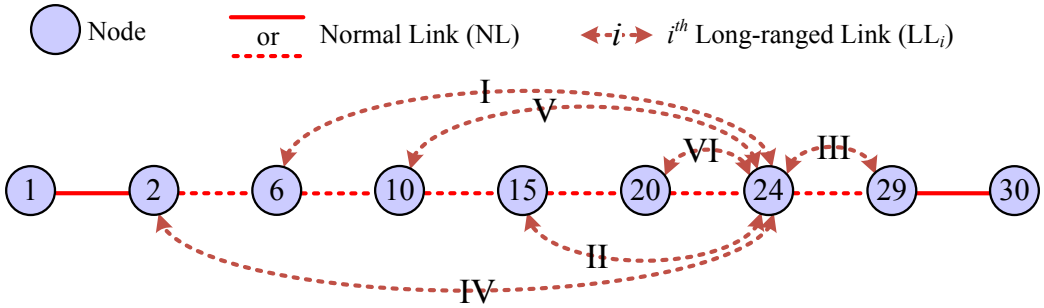


Figure 5.1: The greedy near-optimal LL addition in a 30-node STN. Each LL is marked with a Roman number. The first LL is added between node 6 ($0.2N$) and node 24 ($0.8N$) which are designated as the anchor nodes. It can be seen that node 24, that is, one of the anchor nodes attracts all subsequent LLs and gradually evolves as a hub node. Only relevant nodes are depicted in the figure.

is not a better choice when real-world deployment of new LLs are concerned. In the following, we propose a heuristic approach named sequential deterministic long-ranged link addition (SDLA) to efficiently add k LLs with only $\mathcal{O}(k \times N)$ time.

5.2 LL Addition with SDLA Heuristic

In this section we discuss SDLA based LL addition in an N node undirected and unweighted STN. In a real-world deployment scenario, an STN with N nodes, numbered from 1 to N, can be routers that are positioned as static nodes in a multi-hop communication network and are capable of communicating in multi-channel mode. That is, a set of channels can be used for creating normal links (NLs) with peer nodes, and others are reserved for creating the LLs with distant nodes. All LLs in an STN can be formed by either long-distant wired links or deploying highly directional antennas at the router nodes.

We note from the last section that APL-optimal LL addition with the greedy near-optimal strategy is highly time complex when real-time deployment of LLs is concerned. By contrast, SDLA is a heuristic approach where a set of LLs can be added in a relatively less time. SDLA does not search an LL location exhaustively, in order to get an APL-optimal network, and thus overall time complexity to add an LL is reduced as compared to the greedy near-optimal link addition approach. The SDLA heuristic is designed based on the observations from the greedy near-optimal strategy based LL addition strategy.

Further, the design of SDLA is also influenced by the observations on the fixed fractional locations of anchor nodes in an STN. In the following, the SDLA algorithm is discussed in detail.

5.2.1 The SDLA Algorithm

In order to add k LLs in an N-node STN, we propose SDLA algorithm, presented in Algorithm 5.1. The first LL (LL_1) in Algorithm 5.1 is added based on the simulation and analytical observations [11, 46] on the locations of anchor nodes. The SDLA strategy searches the $0.2N^{th}$ and $0.8N^{th}$ nodes, from node 1 in an N-node STN, to add LL_1 which involves no randomness. LL_1 addition is optimal because it gives the lowest value of APL for the SWSTN. Rest of the sequential LLs are added with the SDLA algorithm by measuring the span distance between the two consecutive LL-node pair. In Figure 5.2, addition of six LLs with the SDLA strategy is shown in a 30-node STN.

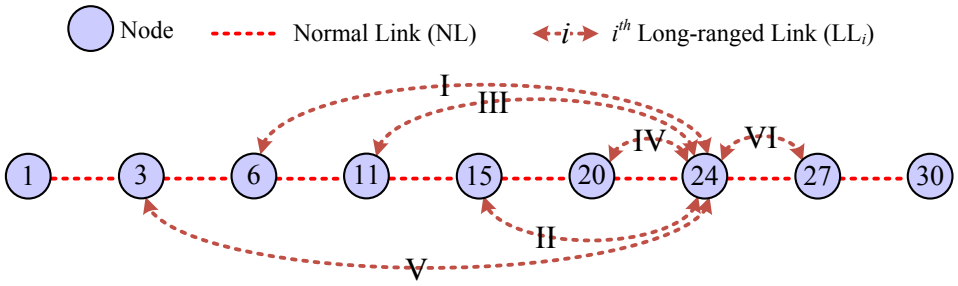


Figure 5.2: Six LLs are added, with SDLA algorithm, in a 30-node STN. First LL (LL_1) is always added between the $0.2N$ and the $0.8N$ nodes in an N-node STN. Here, LL_1 is added between nodes 6 and 24. The $0.8N^{th}$-node, that is, node 24 evolves as the hub node for subsequent LL addition. Rest of the LLs are added based on the maximum span distance. Only relevant nodes are numbered and shown in the figure.

In Figure 5.3, LL_1 is added between the $0.2N^{th}$ and $0.8N^{th}$ nodes. To add the next LL (LL_2) in the STN, the span distance, that is $S_{DIST}(j,\ j+1)$, is measured among node pairs (1, 6), (6, 24), and (24, 30). After comparing all the span distances, SDLA divides the maximum S_{DIST} into two equal parts in order to select the mid-node between j^{th} and $(j+1)^{th}$ LL-nodes as shown in Figure 5.3. It can be seen that the span distance of (6, 24) is the maximum, therefore, to create next LL in the STN, the mid-node, that is node 15, is connected to H. The S_{DIST} estimation is explained in the following.

Algorithm 5.1 Sequential Deterministic LL Addition

Require:

$\mathcal{G} = (\mathcal{V},\ \mathcal{E})$ — A network graph of $\mathcal{V}$ nodes and $\mathcal{E}$ edges
k — Number of long-ranged links (LLs) to be added in $\mathcal{G}$
$(u,\ v)$ — A link between node u and node v
H — The hub node
$SN_{LL(i)}$ — Source node ID for i^{th} LL
$DN_{LL(i)}$ — Destination node ID for i^{th} LL
$\mathbf{S_{DIST}(j,\ j+1)}$ — Span Distance between j^{th} LL-node and $(j+1)^{th}$ LL-node

for $i = 1 \rightarrow k$ **do**
  if $i = 1$ **then**
    $SN_{LL(1)} \leftarrow \lceil 0.2 \times N \rceil^{th}$ Node
    $DN_{LL(1)} \leftarrow \lceil 0.8 \times N \rceil^{th}$ Node
    $H \leftarrow DN_{LL(1)}$
    $(u,\ v) \leftarrow (SN_{LL(1)},\ H)$
    $\mathcal{E} \leftarrow \mathcal{E} \cup (u,\ v)$
  else
    for $j = 1 \rightarrow (N-1)$ **do**
      Find j^{th} and $(j+1)^{th}$ node pair in $\mathcal{G}$ such that
      $\mathbf{S_{DIST}(j,\ j+1)}$ is the **maximum**
      if More than one maximum $\mathbf{S_{DIST}}$ occurs **then**
        Select the maximum $\mathbf{S_{DIST}}$ looking from node 1
      end if
    end for
    $SN_{LL(i)} \leftarrow \left\lceil \frac{SN_{LL(j)} + SN_{LL(j+1)}}{2} \right\rceil^{th}$ Node
    $DN_{LL(i)} \leftarrow H$
    $(u,\ v) \leftarrow (SN_{LL(i)},\ DN_{LL(i)})$
    $\mathcal{E} \leftarrow \mathcal{E} \cup (u,\ v)$
  end if
end for

To add rest of the LLs (after LL_1) in an STN, $0.8N^{th}$ node is considered as the hub node and denoted as H. SDLA calculates the maximum span distance by measuring $S_{DIST}(j,\ j+1)$ to determine the next location of LL addition. S_{DIST} is measured as hop-distance between j^{th} LL-node and $(j+1)^{th}$ LL-node. Here, j^{th} and $(j+1)^{th}$ LL-nodes in S_{DIST} is determined by either of the following methods:

1. j^{th} LL-node and $(j+1)^{th}$ LL-node are consecutive nodes in an STN with different LL connections. Here, j^{th} node is counted from node 1.

2. One of the nodes in j^{th} and $(j+1)^{th}$ node pair is the extreme node without an

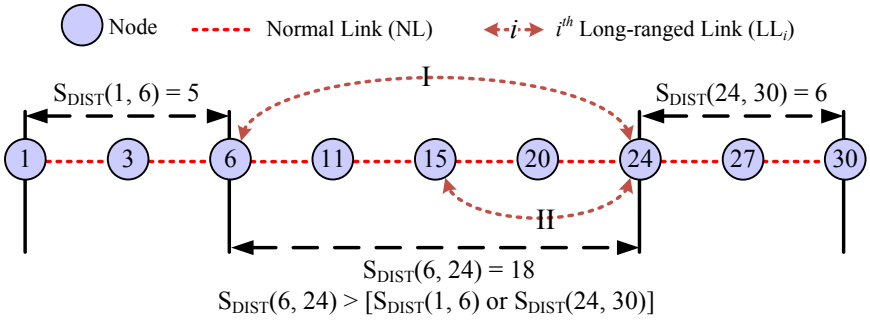


Figure 5.3: Determination of the span distance is explained in a 30-node STN. The first LL (LL_1) is added between nodes 6 and 24. The fractional location of LL_1 is always found to be fixed for different sized STNs. The $0.8N^{th}$ node (in this example, node 24) is designated as the hub node where subsequent LLs are incident. To add the next LL, as shown in the figure, the span distance (S_{DIST}) is calculated among $\{(1,\ 6),\ (6,\ 24),\ (24,\ 30)\}$. As $S_{DIST}(6,\ 24)$ is the largest, it is divided into two equal parts and LL_2 is added between the mid-node of the partitions (here, node 15) and the hub node. Subsequent LLs can also be added using the same strategy. Only relevant nodes are numbered and shown in this figure.

LL (either node 1 or node N), and the other node is the nearest LL connecting node in the STN.

In Figure 5.2, LL_2 to LL_6 are added by estimating the maximum span distance. However, the locations of LLs with SDLA strategy differ from the LL-locations that are found with the near-optimal strategy (see Figure 5.1). This deviation of LL-locations creates only negligible differences in the network properties.

Time Complexity of SDLA Algorithm

The time complexity to add k LLs with the SDLA strategy can be estimated as follows: The first LL is added to an N-node STN in $\mathcal{O}(1)$ time. This is due to the fact that LL_1 is always connected between the anchor nodes, that is, the $0.2N^{th}$ and $0.8N^{th}$ nodes. Rest of the LLs can be added based on the determination of the maximum span distance which can be evaluated in $\mathcal{O}(N)$ time. Therefore, an LL can be added in $\mathcal{O}(1)+\mathcal{O}(N)$ or $\mathcal{O}(N)$ time. Hence, to add k LLs SDLA strategy takes only $\mathcal{O}(k \times N)$ time.

5.3 Performance Analysis of SDLA Algorithm

To understand the network performance after the addition of a few LLs with SDLA, we simulate STNs with varying nodes from 20 to 100. We also compare the performances of SDLA based LL addition and the greedy near-optimal strategy based LL addition in order to quantitatively estimate the efficiency of SDLA to transform an STN to an APL-optimal STN.

5.3.1 Average Path Length Reduction with SDLA

Figure 5.4 shows the APL values, after the addition of six LLs in STNs of varying sizes, with the following strategies: (i) APL values of STNs without LL, (ii) with the random LL addition, (iii) with the near-optimal LL addition, and (iv) with SDLA. It can be seen that the APL value for 100-node STN increases above 30, whereas the random LL addition brings down the APL value nearly to 13. However, the near-optimal and SDLA strategies perform very closely and further bring down the APL about 8.10 to 8.40.

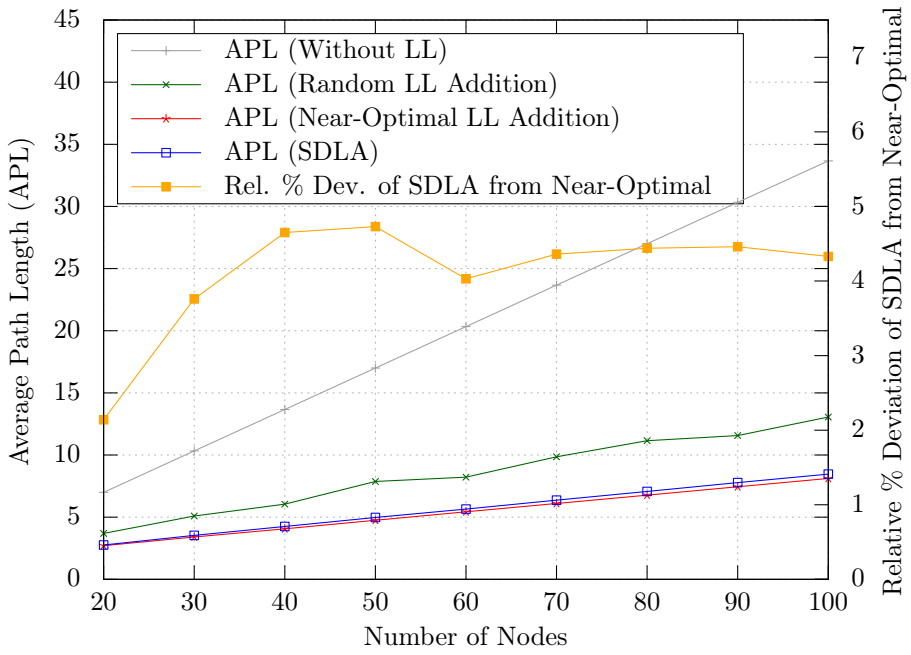


Figure 5.4: Variation of the APL values are shown for different sized STNs for the following cases: (i) APL value before the LL addition, (ii) with the random LL addition (Algorithm 3.1), (iii) with the near-optimal LL addition (Algorithm 3.3), and (iv) with SDLA (Algorithm 5.1). The relative percentage deviation of the APL values of SDLA from the near-optimal LL addition strategy is also shown in the right Y axis.

From the right Y-axis of Figure 5.4, it can be seen that the relative percentage deviation of the APL value, obtained with the SDLA algorithm, is within 6.50% from the near-optimal LL addition algorithm. SDLA allows LL creation in a deterministic manner, thereby, eliminating the random decision based LL addition approach that performs poorly. Further, SDLA is $\mathcal{O}(k \times N)$ time complex than $\mathcal{O}(N^{2k+2} \log\ N)$ for the optimal LL addition and $\mathcal{O}(k \times N^4 \log\ N)$ for the near-optimal LL addition strategies (see Chapter 3), when k LLs are to be added in an N-node STN. The SDLA algorithm is efficient in terms of the time complexity and also deviates negligibly from the near-optimal based LL addition when the objective is to achieve an APL-optimal STN.

Table 5.1
The APL value is calculated in a 50-node STN after the addition of 5, 10, 15, 20, and 25 LLs with the following strategies: (i) random LL addition, (ii) near-optimal LL addition, and (iii) SDLA. It can be seen from the table that the near-optimal and SDLA based LL addition strategies outperform the random LL addition strategy. The percentage deviation of the APL values with SDLA is below 10% (from the table, the maximum deviation is 6.17% with five LLs) from the near-optimal strategy.

Total Number of Added LLs	**Random LL Addition**	**Near-Optimal LL Addition**	**SDLA**	**% Deviation of SDLA from Near-Optimal**
5	8.21	5.50	5.84	**6.17**
10	6.10	3.90	4.13	**6.06**
15	4.82	3.25	3.33	**2.49**
20	4.35	2.90	2.99	**2.89**
25	3.91	2.72	2.81	**3.24**

Table 5.1 demonstrates an example scenario for a 50-node STN, where the APL values are tabulated after various numbers of LL addition with the following strategies: (i) with the random LL addition (all values averaged over 20 seeds), (ii) with the near-optimal LL addition, and (iii) with SDLA strategies. It can be seen from the table that the maximum relative percentage deviation of SDLA from the near-optimal algorithm is 6.17%, whereas the APL values with the random LL addition significantly deviates from the APL values with the near-optimal and SDLA strategies.

5.3.2 Average Clustering Coefficient and Centrality Measures

Average clustering coefficient (ACC) reveals the local connectivity properties of a network (for a detailed discussion on ACC, see Section 2.2.2). As an STN is a sparse network, there is no noticeable variations in the ACC value after the addition of six LLs with the near-optimal and SDLA based LL addition strategies [48]. However, there is a significant change in the centrality measures [23].

Centrality in a network measures the importance of each node. Network centrality (or graph centrality), on the other hand, indexes the tendency of a node to be more central than other nodes. Measure of various graph centralities, such as graph degree centrality (GC_{DC}), graph closeness centrality (GC_{CC}), and graph betweenness centrality (GC_{BC}), give measure of compactness index. All graph centrality values reflect the dominance of a single node with the maximum centrality measure (see Section 2.2.6 for more details on graph centrality).

Table 5.2 shows three centrality measures, (i) with no LL, (ii) with the near-optimal LL addition, and (iii) with SDLA, where the network sizes are varied from 20 nodes to 100 nodes. In the case of 20-node, the deviation of all centrality values with SDLA is high from the near-optimal strategy. In all centrality measures, with the first LL connected in the network, the maximum centrality point [23] is shifted to the hub node which gradually dominates future network evolution by attracting most of the LLs. Therefore, the graph centrality measures for greedy near-optimal and SDLA based strategies are higher compared to the graph centrality values when no LLs are added. Furthermore, it can be seen that the graph centrality values for SDLA (shown in boldface) is comparable to the values obtained from the greedy near-optimal strategy based LL addition. Hence, it can be observed from Table 5.2 that overall performance of the evolved network with SDLA based LL addition strategy is comparable to the exhaustive search based greedy near-optimal LL addition.

Table 5.2
Graph degree centrality (GC_{DC}), graph closeness centrality (GC_{CC}), and graph betweenness centrality (GC_{BC}) measures are shown for various sized STNs. The graph centrality measures are taken for three different cases: (i) with no new LL addition, (ii) six LLs are added with the greedy near-optimal LL addition strategy, and (iii) six LLs are added with the SDLA based strategy. It can be seen that the greedy near-optimal and SDLA based strategies improve the graph centralities compared to the case when no LL is added in the string. Each column values related to SDLA are shown in boldface.

No. of Nodes	*Graph Degree Centrality (GC_{DC})*			*Graph Closeness Centrality (GC_{CC})*			*Graph Betweenness Centrality (GC_{BC})*		
	Without LL	**Near-Optimal LL addition**	**SDLA**	**Without LL**	**Near-Optimal LL addition**	**SDLA**	**Without LL**	**Near-Optimal LL addition**	**SDLA**
20	0.006	0.205	**0.322**	0.087	0.423	**0.512**	0.203	0.504	**0.679**
30	0.002	0.209	**0.209**	0.058	0.431	**0.414**	0.190	0.721	**0.720**
40	0.001	0.155	**0.155**	0.043	0.369	**0.351**	0.184	0.735	**0.724**
50	0.0009	0.123	**0.123**	0.035	0.319	**0.302**	0.180	0.743	**0.736**
60	0.0006	0.102	**0.102**	0.029	0.281	**0.269**	0.178	0.747	**0.741**
70	0.0004	0.087	**0.087**	0.025	0.250	**0.239**	0.176	0.749	**0.747**
80	0.0003	0.076	**0.076**	0.022	0.227	**0.216**	0.175	0.752	**0.746**
90	0.0003	0.068	**0.068**	0.019	0.206	**0.197**	0.174	0.754	**0.750**
100	0.0002	0.061	**0.061**	0.017	0.190	**0.181**	0.173	0.754	**0.751**

5.4 Observations and Discussion

SDLA is an efficient LL addition strategy which can be deployed for network topology design in time critical situations, such as emergency response scenarios and ad hoc deployment of military communication networks. In particular, the APL value reduction with SDLA is deviating less than 10% in our experimental cases, from the near-optimal LL addition strategy. The time complexity of executing SDLA is only $\mathcal{O}(k \times N)$ to add k LLs in an N-node STN. By contrast, the near-optimal decision based LL addition strategy is $\mathcal{O}(k \times N^4 \log\ N)$ time complex, and is not a feasible solution when the size of an STN is very large. Due to the reduced time complexity, SDLA is a better choice for deploying a few LLs in a time critical scenario.

The main concern about SDLA strategy is the traffic fairness issue, as the $0.8N^{th}$ node (the hub node) has to handle higher data traffic. This observation is evident from the point betweenness centrality value of the hub node where most of the traffic in the network is processed. However, in a practical deployment scenario, data carrying capacity of the $0.8N^{th}$ node can be improved easily, because, the exact location of the node in the network is known. By keeping sufficient energy and computing resources at the hub node, overall performance of the network can be enhanced with minimal deployment time complexity. For example, in a military ad hoc STN, the hub node can be a vehicular node capable of carrying many radio/network transceivers and better energy sources.

5.5 Summary

This chapter proposed an efficient LL addition heuristic, sequential deterministic LL addition (SDLA), which can be beneficial for deployment of few LLs in a real-world STN. The SDLA strategy achieves significantly reduced time complexity compared to the near-optimal decision based LL addition strategy. In particular, the complexity of the SDLA algorithm is far superior with $\mathcal{O}(k \times N)$ time than the near-optimal algorithm that takes $\mathcal{O}(k \times N^4 \log N)$ time. Further, the APL deviation with SDLA from the near-optimal strategy is also negligible. SDLA can help efficient design and deployment of moderate sized SWSTNs for applications, such as community computer networks, tactical networks, and emergency response networks.

Chapter 6

Optimal Link Addition in Wireless Sensor Networks

This chapter discusses a real-world application in the context of average path length (APL)-optimal long-ranged link (LL) addition in wireless sensor networks (WSNs). WSNs consist of power constrained sensor nodes that collect raw-data from surveillance and security applications. Sensor nodes are required to send the collected data to the base stations (BSs) using multi-hop relaying for further processing. In many applications, there exists a strict delay deadline on the transfer of data. Addition of a few LLs, that directly connect some sensor nodes to a BS, can achieve small-world properties and, thereby, reduce the delay of transfer to the BS with a corresponding tradeoff in increased transmission power for the LLs. In this chapter, we characterize this tradeoff for the string or linear topology sensor network. We also analytically determine the approximate optimal locations of a single LL as well as two LLs, that can be deployed to the BS in order to achieve the APL to the BS (APLB)-optimal WSNs.

6.1 Existing Literature

WSNs are low-powered, low-cost, power constrained [49] multi-hop wireless relaying networks [50], which are densely deployed for military and wildlife surveillance, habitat monitoring, health monitoring of patients, geological surveying, and smart home applications. The WSN nodes are required to collect and transfer data using multi-hop relaying to a BS under latency constraints. Designing a multi-hop network topology for achieving such latency requirements is challenging and is the motivation for pursuing this piece of work.

In [51], it has been shown that the quality of service requirements on the end-to-end latency in multi-hop networks under low traffic load is equivalent to upper bound constraints on the hop-distance between nodes and the BS. Motivated by this, we consider

the problem of minimizing the APLB, of the WSN nodes, by adding a few LLs to an initial topology of the WSN. However, there also exists transmission power cost over an LL and the APLB has to be traded off with the power cost. We also note that addition of LLs to the network can be viewed as transforming a WSN to a small-world WSN (SWWSN) [1, 2, 3, 4, 32] with the characteristics, such as improved network robustness and reliability, reduction in transmission packet loss, and better routing capability.

Existing approaches to achieve SWWSNs can be classified into two categories: (i) wired LL based approaches, and (ii) wireless LL based approaches. In a large sensor field, with either stationary sensors or sensors with limited mobility, wired LL based approach is a reliable method to create an SWWSN [52]. Conversely, wireless LL based approaches are classified into two categories: (i) heterogeneous SWWSNs and (ii) homogeneous SWWSNs.

Heterogeneous SWWSNs [53, 54, 55] have certain WSN nodes that are more capable than the rest of the WSN nodes to form the LLs. In homogeneous SWWSNs [56], each WSN node is expected to have LL realization capability. Another classification of wireless LLs is based on the physical or virtual nature of the LLs that are formed in an SWWSN [57]. In certain SWWSNs, a physical wireless LL may be formed, and in the rest, a virtual wireless LL may be formed. One example to form a virtual LL based SWWSNs is by using unmanned aerial vehicles, to realize temporary periodic or aperiodic LLs, for creating backbones for gathering data. There exists limited literature on deterministic addition of LLs to transform a WSN to a performance optimized SWWSN (a detailed discussion on the existing deterministic LL addition strategies can be found in Section 3.1.2).

To the best of our knowledge, there exists no analytical approach to identify the locations at which an LL should be added in a WSN to achieve an APL-optimal SWSWN. In this chapter, we analytically identify the optimal locations of a single and two concurrent optimal LLs in a string topology WSN to optimally tradeoff the APLB value and the transmission power in the network.[1] We consider the case of a single BS where all sensor nodes send raw-data over the multi-hop relaying. Our finding leads to the fast LL

[1]Note that our analytical approach is equally applicable when more than two optimal LLs are concerned. However, in this thesis, we restrict our observations only up to two optimal LLs.

deployment algorithms to obtain network topologies with reduced APLB. Achieving the same objective with the exhaustive network simulation has high time complexity [46] and is not a feasible solution for real-world implementation.

6.2 System Model and Link Addition Problem

We consider an N-node WSN, with the nodes indexed by $(1,\ 2, \ldots,\ N)$. The nodes are assumed to have a linear topology, so that the i^{th} node transmits to the $(i+1)^{th}$ node and so on. The N^{th} node is considered to be the BS to which all other nodes are sending data via multi-hop relaying. We also assume that the distance between the i^{th} and $(i+1)^{th}$ nodes is unity. In many real-world scenarios the WSN can be deployed in a linear topology, such as along a highway with uniform distance between the nodes. Since we are interested in the number of hops as a distance measure, such an assumption is reasonable. The APLB of a string topology WSN can be estimated as follows:

$$\frac{1}{N}\sum_{i=0}^{N-1} d(i,\ N), \tag{6.1}$$

where $d(i,\ N)$ is the length of the shortest path between node i and the BS positioned at the N^{th} fractional location in an N-node string topology WSN. We assume that LLs can be added between M pairs of nodes in $(1,\ 2, \ldots,\ N)$ so as to reduce the APLB. When LLs are added, the APLB reduces since $d(i,\ N)$ reduces for some set of nodes i. However, when such links are added then there is additional transmission power that needs to be expended.

Suppose an LL is added between nodes i and j in a linear topology WSN. We model the power expended as being proportional to the square of the distance between the two nodes, that is, the excess power expended in transmission over the LL is $K(i-j)^2$, where K is a positive constant (accounting for the path-loss; fast fading is not considered since the sensors and environment are assumed to be stationary). We assume that the transmitted power is 10 dBm for a 10-hop LL, so that K is 0.1 for a received power level of –20 dBm leading to a bit error rate (BER) of 10^{-3} when the binary frequency shift keying (BFSK) modulation is used. The LL addition problem that we consider here is to

optimally add M LLs to an STN so as to minimize the APLB values subject to a constraint on the excess power in transmission over the LLs. We consider the problem of adding M LLs so as to minimize

$$\sum_{i=0}^{N-1} d(i,\ N) + \lambda K \sum_{m=1}^{M} (i_m - j_m)^2 \tag{6.2}$$

where the m^{th} LL is added from i_m to j_m and λ is interpreted as a positive Lagrange multiplier (we consider $\lambda K \simeq \lambda$). We note that instead of APLB, the total path length term can be considered since N is fixed. The total path length term does not have a closed-form expression in terms of the decisions, that is, the locations i_m and j_m, and hence, the optimization problem of Equation (6.2) is not in an easily solvable form. We propose an approach which obtains an approximate closed-form expression for the total path length in terms of the decisions i_m and j_m using a dense graph approximation. The optimization problem of Equation (6.2) then becomes more amenable to an analytical solution. We present approximate solutions for the case where M=1 and M=2.

6.3 A Dense Graph Approximation: Analytical Solution

We obtain an N-node dense string graph as a limit of a sequence of string graphs, each with nodes $(1,\ 2, \ldots,\ N)$, as $N \to \infty$. We assume that the distance between the nodes i and $i+1$, as well as the link to be added between p_1 and p_2 is $\frac{1}{N}$. We note that as far as the optimization of APL is concerned, scaling all distances in a graph by the same quantity does not matter. The scaling which we have chosen leads to a limit graph consists of a continuum of nodes in the interval [0, 1]. Suppose $d_{sp}(s)$ is the shortest path length of a node at position s to the BS. The total path length f_{sp} for the limit graph is $\int_0^1 d_{sp}(s)ds$. We also note that $d_{sp}(s)$ changes according to the number and positions of the added LLs. We explicitly obtain $d_{sp}(s)$ for two cases below.

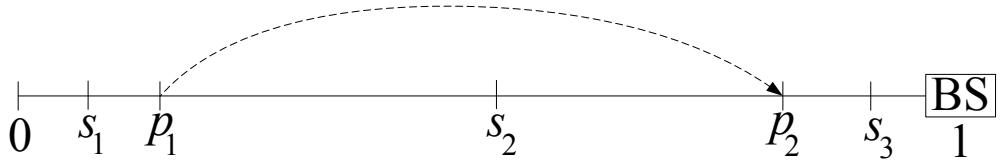


Figure 6.1: A single long-ranged link is optimally added in a string topology wireless sensor network to minimize the objective function in Equation (6.2). There are three cases, based on the location of a source node s (i.e., $s_1,\ s_2$, and s_3), $[0,\ p_1],\ (p_1,\ p_2)$, and $[p_2,\ 1]$ to estimate the shortest path toward the base station.

6.3.1 Case where M=1

Here, we consider the case where only a single LL is added. The LL is added between the nodes at positions p_1 and p_2 (see Figure 6.1) for the dense limit graph. The total path length is a function of p_1 and p_2 and is denoted as $f_{sp}(p_1,\ p_2) = \int_0^1 d_{sp}(s)ds$. It can be seen that $d_{sp}(s)$ depends on whether s is in $[0,\ p_1],\ (p_1,\ p_2)$, or in $[p_2,\ 1]$. If $s \in [0,\ p_1]$, the shortest path from s to 1 is s to p_1, then via the LL $(p_1,\ p_2)$ and then p_2 to 1. Therefore, $d_{sp}(s) = p_1 - s + 1 - p_2$. If $s \in [p_1,\ p_2]$, then $d_{sp}(s) = \min(s - p_1 + 1 - p_2,\ 1 - s)$. With the similar approaches, we obtain that

$$d_{sp}(s) \quad = \quad \begin{cases} p_1 - s + 1 - p_2 & \text{if } s \in [0,\ p_1]\,, \\ s - p_1 + 1 - p_2 & \text{if } s \in \left[p_1,\ \frac{p_1+p_2}{2}\right], \\ 1 - s & \text{if } s \in \left[\frac{p_1+p_2}{2},\ p_2\right] \bigcup [p_2,\ 1]\,. \end{cases}$$

Since $f_{sp}(p_1,\ p_2) = \int_0^1 d_{sp}(s)ds$, substituting the above expressions for $d_{sp}(s)$ we obtain that

$$f_{sp}\,(p_1,\ p_2) = \frac{3}{4}p_1^2 - \frac{1}{2}p_1p_2 - \frac{1}{4}p_2^2 + \frac{1}{2}. \tag{6.3}$$

The power required for transmitting over the LL is

$$f_e(p_1,\ p_2) = \lambda(p_2 - p_1)^2. \tag{6.4}$$

Therefore, the objective function $f(p_1,\ p_2)$ to be minimized for the dense graph is

$$f(p_1,\ p_2) = f_{sp}(p_1,\ p_2) + f_e(p_1,\ p_2), \tag{6.5}$$

and our problem is to

$$\begin{aligned} \text{minimize} \quad & f(p_1,\ p_2) \\ \text{such that} \quad & p_1,\ p_2 \in (0,\ 1), \\ & p_1 < p_2. \end{aligned} \tag{6.6}$$

Equation (6.6) has no extrema for $(p_1,\ p_2)$ in the interior of the constraint set. Therefore, any minima is on the boundary of the constraint region. We note that there exist different possibilities as shown in Figure 6.2.

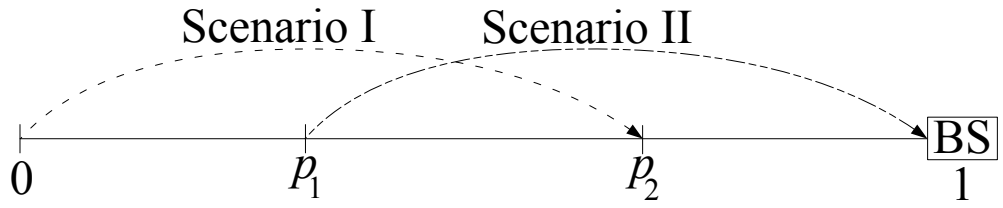


Figure 6.2: Two scenarios to add an optimal long-raged link: (i) (first node, p_2) and (ii) (p_1, BS).

In Figure 6.2, two possible cases are shown: (i) an LL is connected between the first node and p_2, or (ii) the LL is added between p_1 and the BS. To find the optimal location of the first LL, we need to consider each scenario, as depicted in Figure 6.2, separately as described in the following.

Scenario I: LL between (first node, p_2)

We first consider the case where the LL is added between the first node and p_2 (see Scenario I in Figure 6.2). Then we have that $d_{sp}(s)$ depends on whether $s \in [0,\ p_2)$ or $[p_2,\ 1]$. Proceeding as before we obtain that

$$f_{sp}\,(p_2) = \frac{1}{2} - \frac{1}{4}p_2^2. \tag{6.7}$$

By considering the power term, we have that

$$f\left(p_2\right) = \frac{1}{2} - \frac{1}{4}p_2^2 + \lambda p_2^2. \tag{6.8}$$

If $\lambda \geqslant \frac{1}{4}$, the optimal value of p_2 is identified to be 0 and hence, $f(p_2) \simeq \frac{1}{2}$. By contrast, if $0 \leqslant \lambda < \frac{1}{4}$, the optimal value of p_2 becomes 1 and $f(p_2) \simeq \frac{1}{4} + \lambda$.

Scenario II: LL between ($\mathbf{p_1}$, BS)

We again consider the case where the LL is added between p_1 and the BS (see Scenario II in Figure 6.2) where the BS which is situated at the N^{th} location in an N node string WSN ($p_2 = 1$). The evaluation of $d_{sp}(s)$ depends on whether $s \in [0,\ p_1)$ or $[p_1,\ 1]$. Proceeding as before, we obtain that

$$f_{sp}\left(p_1\right) = \frac{1}{4}(p_1 - 1)^2 + \frac{1}{2}p_1^2. \tag{6.9}$$

Similarly, by considering the power expression, we have that

$$f\left(p_1\right) = \frac{1}{4}(p_1 - 1)^2 + \frac{1}{2}p_1^2 + \lambda(1 - p_1)^2. \tag{6.10}$$

Since $\frac{d^2 f(p_1)}{dp_1^2} \geqslant 0$ for all p_1, it is clear that the optimal value of p_1^* minimizing $f(p_1)$ is

$$p_1^* = \frac{1 + 4\lambda}{3 + 4\lambda}. \tag{6.11}$$

The minimum value of $f(p_1)$ is $\frac{1+4\lambda}{2(3+4\lambda)}$. Note that for any value of λ, $f(p_1) < f(p_2)$. The optimal location of a single LL is always between (p_1^*, BS). When there is no power constraint, that is, $\lambda = 0$, p_1^* is identified at 0.33 for the dense limit graph. The approximate solution to Equation (6.2) is that the first LL to be added between nodes n and N such that $\frac{n}{N} \approx 0.33$. We compare our approximate solution (with $\lambda = 0$) to the solution obtained by the greedy optimal search in Table 6.1. The greedy optimal LL addition strategy tries all possible locations of i_1 and j_1 in a string graph for a given N, that is,

$$\min_{\substack{i_1 \in \{1,\ 2,\ldots,\ N-1\} \\ j_1 \in \{i_1+1,\ldots,\ N\}}} \left(\sum_{i=0}^{N-1} d(i,\ N) + \lambda (i_1 - j_1)^2 \right),$$

in order to obtain the minima. We find that our approximate solution yields the same value of APLB for a number of values of N (the mismatch in the location depends on how n, such that $\frac{n}{N} \approx 0.33$, is chosen). Assumptions that are considered for simulation of the greedy optimal decision based LL addition are as follows: An N-node STN is taken and the N^{th} node is assumed to be the BS. An optimal LL is then added between a node pair based on the greedy optimal LL addition algorithm (see Algorithm 3.2 in Section 3.3.1). Note that we show, in Table 6.1, the APLB value after a single LL addition with the approximate solution as well as with the greedy optimal LL addition strategy. We observe from the table that our approximate solution efficiently finds optimal location of a single LL in the string topology SWWSN.

Table 6.1
Location of a single optimal long-ranged link: analytical and simulation observations. Note that we consider no power constraint ($\lambda = 0$) in this observation.

No. of Nodes	Analytical Observations		Simulation Observations	
	Location of Single LL	APLB	Location of Single LL	APLB
10	(4, 10)	2.33	(3, 10)	2.33
20	(7, 20)	4.00	(6, 20)	4.00
30	(10, 30)	5.66	(10, 30)	5.66
40	(14, 40)	7.33	(13, 40)	7.33
50	(17, 50)	9.00	(16, 50)	9.00

Tradeoff of APLB with Power

The location of LL for $M{=}1$ with one end at the BS, are functions of λ which is the weight factor for the LL transmission power. There exists a natural tradeoff between the APLB values and the amount of transmission power which can be obtained by varying λ. From Equation (6.4) we have that the length of an LL is directly proportional to the required

transmission power to deliver a data packet. To minimize the transmission power, the length of an LL should be small. Conversely, to minimize the end-to-end hop distance between a sensor node and the BS, APLB has to be reduced, and thus, the LL length should be large. This tradeoff in APLB and transmission power is shown in Figure 6.3. For calculating the transmission power, we assume a minimum received power of –20 dBm for a BER of 10^{-3} (with BFSK) at the BS [58]. The approximate solution is estimated by identifying the LL locations, with varying λ, from analytical solution and then directly provide the LL locations to the simulation experiment. It can be observed from the figure that the tradeoff curve achieves through the analytical solution always shows the lower bound responses. We also observe that the tradeoff curves of the simulation and the approximate solutions are exactly matching for different values of N. Further, from Figure 6.3, it can be observed that one end of the relative fractional locations of a single LL is moving toward the BS when minimizing the transmission power is concerned.

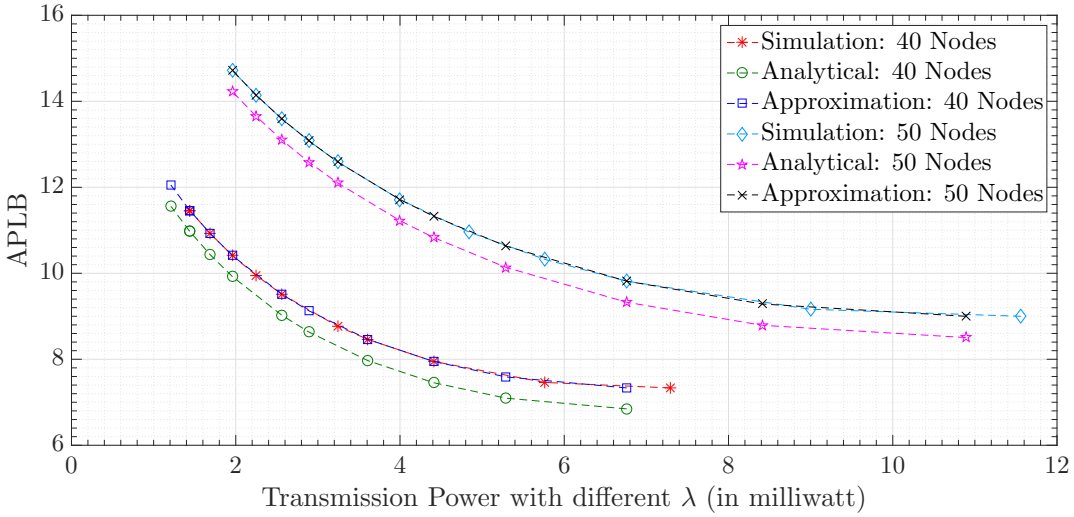


Figure 6.3: Tradeoff of APLB and the transmission power, for 40- and 50-node string topology wireless sensor networks (see legends), with a *single optimal LL* (M=1): Minimum received power is assumed to be –20 dBm and BFSK modulation with minimum BER of 10^{-3}, where λ ranges from 0 to 1.

6.3.2 Case where M=2

We consider the case for M=2, that is, two LLs are added to the string topology WSN. In order to obtain the locations analytically, we make a simplifying assumption that one end of both the LLs are connected to the BS. The theoretical motivation for this simplification

is that for $M{=}1$ we have obtained that one end of the single LL is connected to the BS. We also note that in practical scenarios, the BS in a WSN can be equipped with more energy resources and higher computing power. It is reasonable to have the LLs connected to the BS.

We assume that the two optimal LLs are connected between (p_1, BS) and (p_2, BS), where $p_1 < p_2$. Note that this assumption is only needed to analytically identify the LL connecting locations in the limit graph. The shortest path is again a function of p_1 and p_2, that is, $f_{sp}(p_1,\ p_2) = \int_0^1 d_{sp}(s)ds$, and $d_{sp}(s)$ can be obtained based on the location of s in $[0,\ p_1]$, $(p_1,\ p_2)$, or $[p_2,\ 1]$ as follows:

$$d_{sp}(s) \quad = \quad \begin{cases} p_1 - s & \text{if } s \in [0,\ p_1]\,, \\ s - p_1 & \text{if } s \in \left[p_1,\ \frac{p_1+p_2}{2}\right], \\ p_2 - s & \text{if } s \in \left[\frac{p_1+p_2}{2},\ p_2\right], \\ s - p_2 & \text{if } s \in \left[p_2,\ \frac{p_2+1}{2}\right], \\ 1 - s & \text{if } s \in \left[\frac{p_2+1}{2},\ 1\right]. \end{cases}$$

Using the method similar to Section 6.3.1, we get that

$$f_{sp}\left(p_1,\ p_2\right) = \frac{1}{2}p_1^2 + \frac{1}{4}\left(p_1 - p_2\right)^2 + \frac{1}{4}\left(p_2 - 1\right)^2. \tag{6.12}$$

Therefore, after adding the power optimization terms, we get that

$$f\left(p_1,\ p_2\right) = \frac{1}{2}p_1^2 + \frac{1}{4}\left(p_1 - p_2\right)^2 + \frac{1}{4}\left(p_2 - 1\right)^2 + \lambda(1-p_1)^2 + \lambda(1-p_2)^2. \tag{6.13}$$

We check whether $f(p_1,\ p_2)$ is convex by considering the determinant of the Hessian (H) of Equation (6.13)

$$det(H) = \left| \begin{bmatrix} \frac{3}{2} + 2\lambda & -\frac{1}{2} \\ -\frac{1}{2} & 1 + 2\lambda \end{bmatrix} \right| \geqslant 0.$$

Since H is positive definite, Equation (6.13) is strictly convex and if $p_1^* \in [0,\ 1]$ and $p_2^* \in [0,\ 1]$ can be found satisfying $\frac{\partial f(p_1,p_2)}{\partial p_1} = 0$ and $\frac{\partial f(p_1,\ p_2)}{\partial p2} = 0$, then p_1^* and p_2^* are the minima. We obtain that

$$p_1^* = \frac{2\lambda + 0.5p_2}{1.5 + 2\lambda}, \qquad (6.14)$$
$$p_2^* = \frac{(0.5 + 2\lambda)(3 + \lambda) + 2\lambda}{(1 + 2\lambda)(3 + \lambda) - 0.5},$$

and the minimum value of $f(p_1,\ p_2)$ is

$$\frac{512\lambda^6 + 3840\lambda^5 + 10112\lambda^4 + 10136\lambda^3 + 4321\lambda^2 + 744\lambda + 90}{4(16\lambda^3 + 68\lambda^2 + 62\lambda + 15)^2}.$$

Table 6.2
Locations of two optimal long-ranged links: analytical and simulation observations. Note that we consider no power constraint ($\lambda = 0$) in this observation.

No. of Nodes	Analytical Observations		Simulation Observations	
	Locations Two LLs	APLB	Locations of Two LLs	APLB
10	(2, 10), (6, 10)	1.78	(2, 10), (5, 10)	1.78
20	(4, 20), (12, 20)	2.79	(4, 20), (11, 20)	2.79
30	(6, 30), (18, 30)	3.79	(6, 30), (17, 30)	3.79
40	(8, 40), (24, 40)	4.79	(8, 40), (23, 40)	4.79
50	(10, 50), (30, 50)	5.80	(10, 50), (29, 50)	5.80

If $\lambda = 0$, then p_1^* and p_2^* are 0.2 and 0.6, respectively. For N-node graph, we add two LLs $(n_1,\ N)$ and $(n_2,\ N)$ such that $\frac{n_1}{N} \approx 0.2$ and $\frac{n_2}{N} \approx 0.6$. A similar procedure can be used for any λ. We compare our approximate solution for M=2, with no power constraint, to the solution obtained by the greedy optimal LL addition strategy in Table 6.2. We find that the approximate solution yields the same value of APLB for different values of N.

Tradeoff of APLB with Power

The locations of the LLs for M=2 with one end at the BS, are also functions of λ. The tradeoff in the APLB and transmission power, with M=2, is shown in Figure 6.4. To calculate the transmission power, similarly as M=1, we assume a minimum received power of –20 dBm for a BER of 10^{-3} (with BFSK) at the BS. It can be seen from Figure 6.4 that

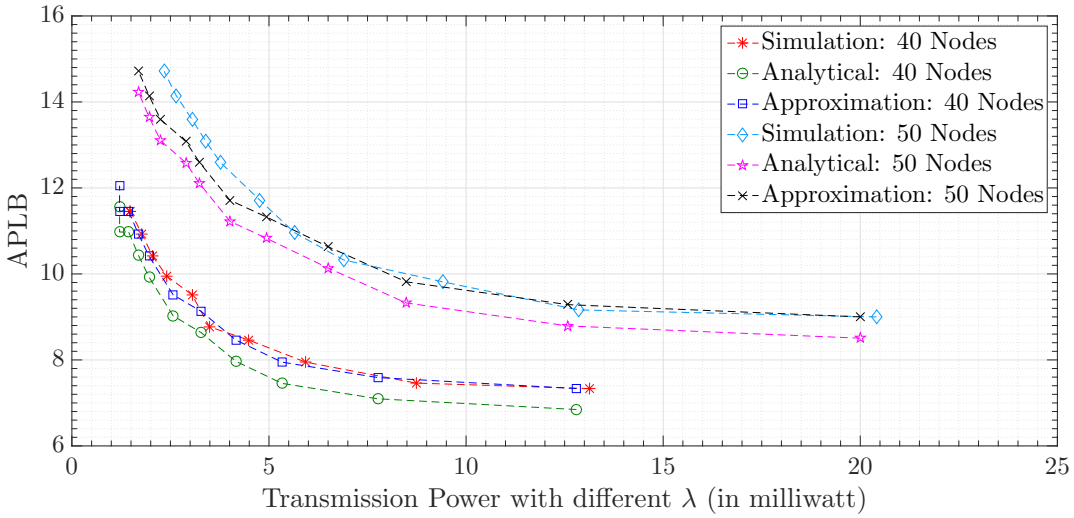


Figure 6.4: Tradeoff of APLB and the transmission power, for 40- and 50-node string topology wireless sensor networks (see legends), with the *two optimal LLs* (M=2): Minimum received power is assumed to be –20 dBm and BFSK modulation with minimum BER of 10^{-3}, where λ ranges from 0 to 1.

the tradeoff curve achieved through the analytical solution always shows the lower bound responses. Further, it can also be observed that the tradeoff curves of the simulation and approximate solutions are within 10% for different values of N. Moreover, similarly as M=1, the relative fractional locations of one end of the optimal LLs are also changing (by keeping other end fixed at the BS) with different values of transmission power.

6.4 Summary

In this chapter we analytically determined the approximate locations of a single and two optimal LLs when the BS of a WSN is situated at one end in a finite sized STN. We observed that one end of an optimal LL is always connected to the BS. Such an addition is feasible as the BS can be equipped with more energy resources and computing power. The optimal fractional locations of a single LL is $(0.33,\ BS)$ and two LLs are $(0.2,\ BS)$ and $(0.6,\ BS)$ when there is no constraint on the transmission power. The effect of increase in the transmission power is also studied after the addition of a single or two LLs. The design of an APL-optimal SWWSN is beneficial for the efficient data transmission and enhancing the longevity of the sensor nodes.

Chapter 7

Achieving Capacity-Enhanced Small-World Networks

Until now, we considered deterministic LL addition to minimize the end-to-end hop distance in a network and to incorporate small-world characteristics. However, a real-world network, whether it is communication or transportation, can suffer from degradation in average network flow capacity (ANFC) due to the presence of one or more bottleneck nodes. In this chapter, an exhaustive search based LL addition algorithm, maximum flow capacity (MaxCap), which deterministically maximizes ANFC based on the maximum flow between node pairs in the context of weighted undirected networks is presented. Further, based on the observations from MaxCap, an LL addition heuristic, average flow capacity enhancement using small-world characteristics (ACES), which significantly enhances ANFC and the length-type product (LTP) of a link in a weighted undirected network is proposed. The chapter discusses performance of ACES through exhaustive simulations on various arbitrary and real-world road networks. Further, we compare MaxCap and ACES performances, with existing deterministic LL addition strategies such as near-optimal-, maximum betweenness centrality-, and maximum closeness centrality based LL addition, in the context of maximizing ANFC values.

7.1 Existing Literature

Network flow capacity (NFC),[1] which is defined as the maximum amount of information or the highest number of objects (e.g., vehicles) that can be transmitted (or transported) over a network in unit time, is one of the most critical parameters while designing an efficient communication or transportation network. However, a network with large end-to-end hop distance (EHD) results in higher transmission delay due to the presence of several intermediate bottleneck nodes. Deploying a few LLs can efficiently reduce EHDs

[1] In this chapter, *network flow capacity* and *network capacity* are used interchangeably.

in a regular network and thus, overall flow capacity of the network can be improved. With the creation of a few LLs, a regular network is gradually transformed to a small-world (SW) network [2, 3]. The SW characteristics in a network can be defined by lower average path length (APL) along with low to moderate value of average clustering coefficient (ACC) [30].

There exist many real-world scenarios, such as road networks, communication networks, and transportation networks where NFC is one of the crucial parameters. For example, a few flyovers can be constructed in road networks [59] to avoid traffic congestion. However, without proper identification of the flyover locations, a road network may not get benefited, when traffic fairness and flow capacity enhancement are concerned.

To the best of our knowledge, NFC enhancement has been considered only in a handful of literature. In [60], authors considered data model as well as graph model for designing an efficient hierarchical road network in order to better organize the route planning in vehicle navigation system. Here, data model represents hierarchical road networks and graph model mimics multi-layered model of road networks. On the other hand, authors in [61] developed a hierarchical community detection algorithm which incorporates within-community as well as between-community routing strategies, and also supports effective route computation on large road networks.

In [62], based on the cell transmission model of freeway traffic flow, a macroscopic traffic flow model was developed to analyze dynamical systems in the context of equilibrium flow and convergence. Furthermore, based on the observations, authors proposed an optimal ramp metering strategy to maximize network throughput. In [63], a vehicle rerouting strategy was discussed to avoid congestion due to some unexpected events such as road accidents. Moreover, the proposed strategy helps in achieving average lower travel time while guaranteeing higher travel time reliability when rerouting in a congested network is concerned. Recently, authors in [64] proposed a set of strategies to identify optimal locations of check-in nodes (where services such as refueling can be carried out) in a real-world road network to maximize overall profit in the network.

It can be seen that none of the existing literature implements the small-world characteristics to enhance the NFC value. We propose an exhaustive search based LL addition

algorithm, maximum flow capacity (MaxCap), to deterministically maximize the average NFC (ANFC) of a weighted undirected network[2] that cannot be achieved with existing random and deterministic LL addition strategies. Moreover, based on the observations from MaxCap based LL addition algorithm, we propose a heuristic, average flow capacity enhancement using small-world characteristics (ACES), which transforms a regular network to a capacity-enhanced small-world network by adding a few LLs.

In the following, we discuss average network flow capacity which will be used later in this chapter to explain the performances of MaxCap and ACES based LL addition strategies.

7.2 Average Network Flow Capacity

To identify the maximum flow capacity (FC) between a node pair in a network graph $\mathcal{G}$, in this chapter, we use the max-flow min-cut theorem [65]. Network flow capacity (NFC), on the other hand, can be estimated by summing maximum FCs among all possible node pairs as follows:

$$NFC = \sum_{i \neq j} FC_{ij}. \tag{7.1}$$

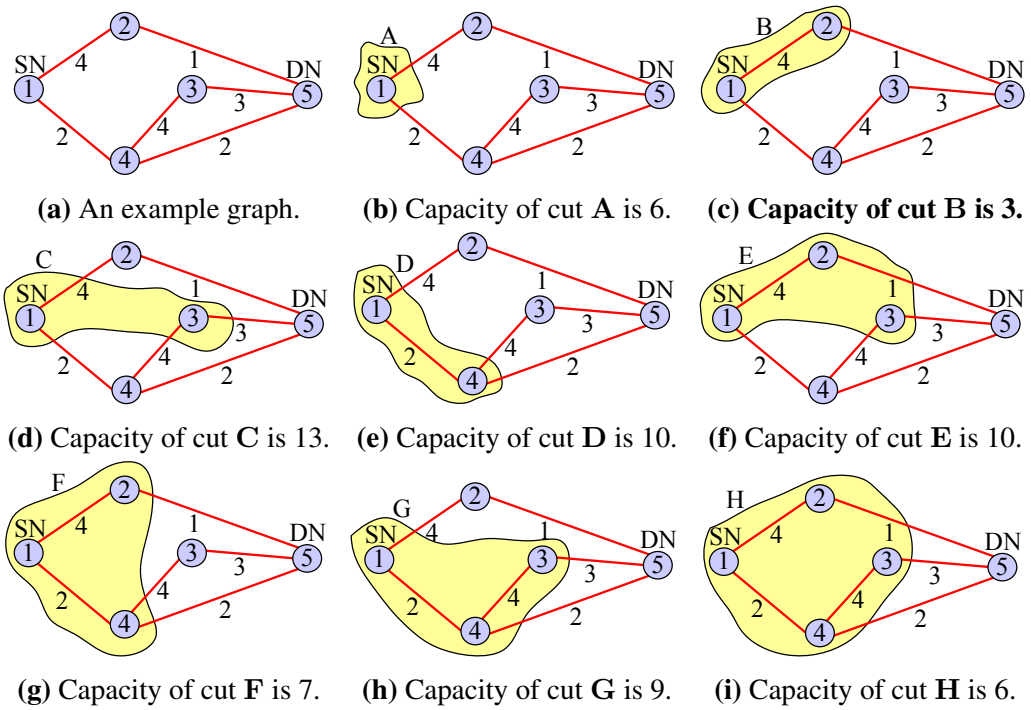
In Equation (7.1), FC_{ij} represents the maximum flow capacity between a node pair $(i,\ j)$. Before explaining ANFC of a network, we briefly discuss max-flow min-cut theorem.

7.2.1 Max-Flow Min-Cut Theorem

The max-flow min-cut theorem finds the maximum flow capacity of a network. A cut is a partition of network nodes into two sets such that the source node is in one set and the destination node is in the other set. Max-flow min-cut theorem [65] states that the maximum value of a flow between a source node (SN) and destination node (DN) pair

[2]In this chapter, we consider only undirected network with positive weights. Note that an unweighted network is a special case where link weights are assumed to be unity.

can be evaluated as the minimum value of capacities of all cuts that separates the SN from the DN in a network. In order to understand the max-flow min-cut theorem, an example of 5-node undirected weighted network is shown in Figure 7.1.



(a) An example graph. **(b)** Capacity of cut **A** is 6. **(c) Capacity of cut B is 3.**

(d) Capacity of cut **C** is 13. **(e)** Capacity of cut **D** is 10. **(f)** Capacity of cut **E** is 10.

(g) Capacity of cut **F** is 7. **(h)** Capacity of cut **G** is 9. **(i)** Capacity of cut **H** is 6.

Figure 7.1: An example 5-node weighted undirected graph and its all possible cuts between {SN, DN}. In (a), an example 5-node weighted undirected network graph is shown. From (b)-(i), all possible min-cuts between node 1 (source node) and node 5 (destination node) are given. From all possible min-cuts, the cut of (c) (i.e., the cut **B**) is the lowest with value 3. Therefore, the maximum flow capacity between nodes 1 and 5 is evaluated to be 3 units.

In Figure 7.1(a), weight of each link is scripted which represents the maximum link capacity in the network. In this example, SN is node 1 and DN is node 5. When a cut separates the SN from the DN, maximum flow capacity of the cut is evaluated by summing all outgoing link capacities from the cut. In Figure 7.1(a), all links are assumed to be undirected, therefore, weights of all links that are coming out from the cut are to be added in order to get the maximum flow capacity of the given cut.

Let $\mathbf{P}$ be all possible sets of nodes containing node 1 (i.e., the SN) and $\bar{\mathbf{P}}$ be all possible set of nodes with node 5 (i.e., the DN). Then, the capacity of each cut is calculated as the sum of link capacities present between nodes of $\mathbf{P}$ and nodes of $\bar{\mathbf{P}}$. Note that only $2^{(N-2)}$ sets are possible in an N-node network. Figures 7.1(b)-(i) show all possible cuts

Table 7.1
All possible flow capacities between a source node and a destination node. Here, all possible flow capacities are listed between nodes 1 (SN) and 5 (DN) of Figure 7.1(a). Note that cuts identified by $\mathbf{P}$ and $\overline{\mathbf{P}}$ and their capacities are listed in the table.

Cut ID	Node Set P	Node Set $\overline{\mathbf{P}}$	Capacity
A	$\{1\}$	$\{2,\ 3,\ 4,\ 5\}$	6
B	$\{1,\ 2\}$	$\{3,\ 4,\ 5\}$	3
C	$\{1,\ 3\}$	$\{2,\ 4,\ 5\}$	13
D	$\{1,\ 4\}$	$\{2,\ 3,\ 5\}$	10
E	$\{1,\ 2,\ 3\}$	$\{4,\ 5\}$	10
F	$\{1,\ 2,\ 4\}$	$\{3,\ 5\}$	7
G	$\{1,\ 3,\ 4\}$	$\{2,\ 5\}$	9
H	$\{1,\ 2,\ 3,\ 4\}$	$\{5\}$	6

between nodes 1 (i.e., SN) and 5 (i.e., DN) and corresponding flow capacity values are also noted which can be found in Table 7.1.

For example, Figure 7.1(b) shows a cut, where $\mathbf{P}=\{1\}$ and the corresponding capacity is (4 + 2) or 6 units. Similarly, for another cut, $\mathbf{P}=\{1,\ 3\}$, the capacity is (4 + 2 + 4 + 3) or 13 units (see Figure 7.1(d)).

In Table 7.1, there are eight possible cuts out of which the cut with $\mathbf{P}=\{1,\ 2\}$ has the lowest value of capacity. Therefore, by max-flow min-cut theorem [65], maximum possible capacity between node 1 and node 5 is 3 units (see Figure 7.1(c)). Therefore, if a link with weight 3 units can be added between node 1 and node 3, maximum capacity of the network (with SN-DN pair as $\{1,\ 5\}$) enhances to 6 units as shown in Figure 7.2. Note that, in this example scenario, we have added an LL of weight 3 units because next min-cut capacity cannot reach beyond 6 units.

Note further that the network APL has become smaller than the original network when a new link between nodes 1 and 3 is added. Therefore, small-world network creation can also be carried out with the dual objectives of network capacity enhancement and APL reduction.

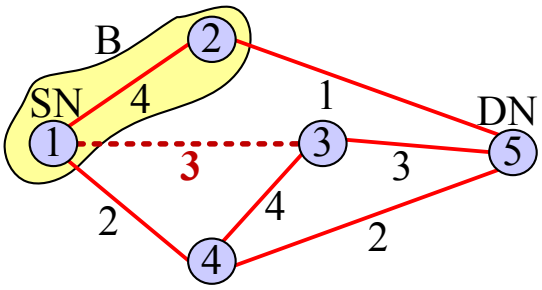


Figure 7.2: Addition of one link of 3 units between node 1 and node 3 can increase the flow capacity to 6 units between {SN, DN}. The newly added link is showing with a *dashed* line.

Why to Consider Average NFC (ANFC)?

In the discussion of the max-flow min-cut theorem, procedure of calculating maximum flow capacity between a pair of nodes is explained. Now, to calculate the network flow capacity (NFC) of a network, it is required to add maximum flow capacity values among all possible node pairs in the network. NFC is a cumulative sum of all possible node pairs. The average NFC (ANFC) of an N-node network can be expressed as follows:

$$ANFC = \frac{NFC}{{}^{N}C_2}. \tag{7.2}$$

ANFC is considered over a certain *flow capacity* due to the fact that the LL addition strategy based on maximizing flow capacity frequently finds pendant nodes[3] in a network to add an LL. This is because of the limitation imposed by the max-flow based capacity evaluation (see Section 7.2.1), where the maximum possible network capacity of a network is the minimum cut present in the network [65]. Hence, the choice may not always be helpful and practical when real-world networks are concerned. ANFC helps in efficient network design by avoiding the selection of pendant nodes which do not contribute significantly enhancing the flow capacity of the entire network.

[3] A *pendant node* is a node with a single link (i.e., the nodal degree is 1) in a network.

7.3 Link Addition with Maximum Flow Capacity

Maximum flow capacity (MaxCap) algorithm[4] deterministically maximizes the ANFC value in a weighed undirected network. MaxCap algorithm, at each step, adds an LL in order to maximize ANFC of a network. MaxCap is described in Algorithm 7.1.

Algorithm 7.1 Maximum Flow Capacity Algorithm

Require:

$\mathcal{G} = (N,\ \mathcal{E},\ \mathcal{C})$ — A network with N nodes and $\mathcal{E}$ links with link capacity of $\mathcal{C}$
$(i,\ j)$ — A link exists between i and j
$ANFC$ — Average network flow capacity of $\mathcal{G}$
$\mathcal{C}_{Var}$ — Variable capacity value of an LL
$\mathcal{C}_{Fix}$ — Fixed capacity value of an LL
max_cap — Maximum value of ANFC
max_cap_LL — LL location that returns max_cap
k — Number of long-ranged links (LLs) to be added in $\mathcal{G}$
Initialization: $max\_cap \leftarrow 0$ and $max\_cap\_LL \leftarrow \varnothing$

```
for i = 1 → k do
    for p = 1 → N − 1 do
        for q = p → N do
            if (p, q) ∉ E then
                Add an LL between (p, q) node pair in G
                Assign C_Var or C_Fix                    // Based on the capacity assignment
                Calculate ANFC
                if ANFC > max_cap then
                    max_cap ← ANFC
                    max_cap_LL ← (p, q)
                end if
                Remove the LL
            end if
        end for
    end for
    Add i^th LL between the node pair in max_cap_LL
    Assign corresponding C_Var or C_Fix
    Update network graph G
    max_cap ← 0
    max_cap_LL ← ∅
end for
```

For each LL addition, Algorithm 7.1 searches for all possible node pairs to add an

[4]Rest of the chapter exercises the following naming conventions: *MaxCap based LL addition strategy* and *MaxCap* are synonymous and used interchangeably.

LL. Note that, the algorithm does not consider any parallel link or self loop. Algorithm 7.1 initializes max_cap which holds the maximum value of ANFC in a network and max_cap_LL which holds the corresponding LL location. The MaxCap algorithm is explained in the following.

To select an LL location that results in maximum ANFC value, Algorithm 7.1 exhaustively searches for all possible LL locations in a network (lines 2-15). In order to assign corresponding link flow capacity to the LL added (lines 5-6), two different strategies are followed:

(I) In case of variable link capacity assignment to an LL added, the value of $\mathcal{C}_{Var}$ is set to be the maximum of the flow capacity values of all links that are connected to either of the end nodes of the LL.

(II) On the other hand, value of link capacity is fixed to a certain value for all LLs when link capacity assignment based on $\mathcal{C}_{Fix}$ is concerned.

After assigning link capacity to the LL added, ANFC is calculated with Edmond-Karp algorithm [66] and checks whether it is larger than the already existing max_cap value. If so, the max_cap value is updated with the current ANFC value and corresponding LL location is stored in max_cap_LL (lines 7-11). After selecting the LL that returns maximum ANFC, corresponding link capacity (i.e., $\mathcal{C}_{Var}$ or $\mathcal{C}_{Fix}$ based on the capacity assignment strategy) is assigned (lines 16-18). After this step, max_cap and max_cap_LL are again initialized to zero and null, respectively, to search the next deterministic LL in the network (lines 19-20). Algorithm 7.1 continues searching for the next capacity maximized deterministic LL until k LLs are added to the network. Figure 7.3 shows three example road networks where 10 LLs are added, each with variable flow capacity, based on MaxCap based LL addition.

7.3.1 Time Complexity of MaxCap Algorithm

In order to estimate the time complexity of MaxCap strategy to add k LLs in an N-node network, time complexity to estimate the flow capacity between a node pair needs to be identified. Here, the flow capacity between a node pair is measured using Edmond-Karp

algorithm [66], in $\mathcal{O}(N^3)$ time, at each step of the possible LL location. Hence, to check all LL possibilities, MaxCap algorithm takes $\mathcal{O}(N^2 \times N^3)$ time to identify an LL which gives maximum ANFC value for the network. Therefore, to add k LLs, Algorithm 7.1 is $\mathcal{O}(k \times N^5)$ time complex.

(a) Delhi. **(b)** London. **(c)** Manhattan.

Figure 7.3: Three different road networks including (a) Delhi, (b) London, and (c) Manhattan, where 10 LLs are added with MaxCap strategy with variable flow capacity values. Note that the *dark colored* lines are the LLs added in the networks with MaxCap.

7.4 Average Flow Capacity Enhancement using Small-World Characteristics

From the previous section, it can be seen that MaxCap takes significant amount of time to decide the location of an LL to maximize ANFC. To overcome the shortcoming of MaxCap we propose a heuristic, average flow capacity enhancement using small-world characteristics (ACES), based on the observations from the locations of LLs which are added with MaxCap strategy. ACES strategy is discussed in Algorithm 7.2.

In Algorithm 7.2, min_degree represents minimum degree of a node under consideration. ACES starts with min_degree=1. Hence, the algorithm searches for all nodes with degree 1 and store them in $node\_set$. Moreover, ACES also stores those nodes which are r-hop away from the nodes with min_degree (lines 2-7). When the search is complete, the algorithm identifies whether the total number of LL possibilities using the nodes in the $node\_set$ is sufficient to create k LLs (lines 8-11). If k LL creation possibilities exist, ACES calculates the Euclidean distances between all possible node pairs in the $node\_set$ and k LLs are then added between node pairs (with corresponding $\mathcal{C}_{Var}$ or $\mathcal{C}_{Fix}$ values) in the descending order of the Euclidean distances (lines 12-15).

Algorithm 7.2 Average Flow Capacity Enhancement using Small-world Characteristics

Require:

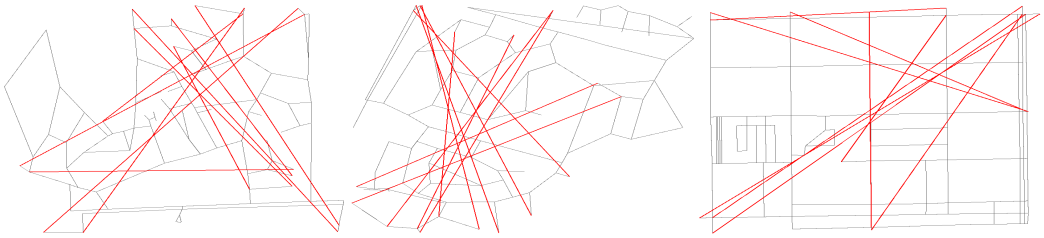
$\mathcal{G} = (N,\ \mathcal{E},\ \mathcal{C})$ — A network with N nodes and $\mathcal{E}$ links with link capacity of $\mathcal{C}$
min_degree — Minimum degree of a node in $\mathcal{G}$
$NE_{i(r)}$ — r-hop neighbors of node i
$D(i)$ — Degree of node i
$node\_set$ — Set of nodes for possible LL creation
$\mathcal{C}_{Var}$ — Variable capacity value of an LL
$\mathcal{C}_{Fix}$ — Fixed capacity value of an LL
k — Number of long-ranged links (LLs) to be added in $\mathcal{G}$
Initialization: r, $node\_set \leftarrow \varnothing$ and $min\_degree \leftarrow 0$

1: $min\_degree \leftarrow min\_degree + 1$
2: **for** $p = 1 \rightarrow N$ **do**
3: **if** $D(p) = min\_degree$ **then**
4: $node\_set \leftarrow node\_set \cup p$ *// Add node with min_degree*
5: $node\_set \leftarrow node\_set \cup NE_{p(r)}$ *// Add r-hop neighbors*
6: **end if**
7: **end for**
8: $count \leftarrow |node\_set|$ *// Cardinality value of $node\_set$*
9: **if** $|^{count}C_2 - \mathcal{E}| < k$ **then** *// $|^{count}C_2 - \mathcal{E}|$ to discard existing links*
10: **go to** 1
11: **end if**
12: Calculate the Euclidean distance between node pairs in $node\_set$
13: Sort node pairs in descending order based on the Euclidean distance
14: Add k LLs among first k sorted node pairs in $node\_set$
15: Assign $\mathcal{C}_{Var}$ or $\mathcal{C}_{Fix}$ to each LL according to the strategy

On the other hand, if creation of k LLs is not possible between the possible node pairs of $node\_set$, Algorithm 7.2 increments the min_degree value by one, and enhance the LL connection possibilities by adding more nodes in the $node\_set$ (lines 8-11). Figure 7.4 depicts three example road networks where 10 LLs are added based on ACES with variable flow capacity assignment to each LL.

7.4.1 Time Complexity of ACES

The time complexity of ACES algorithm (i.e., Algorithm 7.2) is evaluated as follows: Identifying nodes for the set min_degree, to add possible LLs, takes $\mathcal{O}(N)$ time in an N-node network (lines 2-7). To get first k LL locations, sorting operation is performed on the set of Euclidean distances between possible LL connecting node pairs. Hence,



(a) Delhi. **(b)** London. **(c)** Manhattan.

Figure 7.4: Three different road networks including (a) Delhi, (b) London, and (c) Manhattan, where 10 LLs are added with ACES strategy with variable flow capacity for each LL. Note that the *dark colored* lines are the LLs added in the networks with ACES.

first k LL locations, from the sorting operation in descending order, can be found in $\mathcal{O}(k \times N^2)$ time (lines 13-14). All other operations in Algorithm 7.2 can be executed in $\mathcal{O}(1)$ time. Therefore, the time complexity to add k LLs with the ACES strategy is $[\mathcal{O}(N) + \mathcal{O}(k \times N^2) + \mathcal{O}(1)]$ or $\mathcal{O}(k \times N^2)$.

7.4.2 Why does ACES Work?

The heuristic design of ACES strategy is primarily based on the observations from the MaxCap algorithm where an LL is added to maximize the ANFC value of a real-world network. It can be observed that, from Figure 7.3, most of the LLs find pendant nodes or their neighbors as one of the connecting points. In particular, both MaxCap and ACES can successfully identify the importance of a few pendant nodes to significantly enhance the ANFC values of real-world road networks. However, with MaxCap, some LLs are also added between nodes which are geographically nearer.

ACES, on the other hand, identifies all pendant nodes and their r-hop neighbors and then connect an LL between the node pair with the highest geographical distance as can be seen in Figure 7.4. The concept of r-hop neighbors also helps ACES to efficiently search LL locations in a network.

7.5 Performance Evaluation of ACES

In the following, quantitative comparison results are shown to estimate the ACES performance with respect to other deterministic LL addition strategies such as MaxCap, the

near-optimal decision based LL addition to minimize network APL,[5] maximum betweenness centrality (MaxBC), and maximum closeness centrality (MaxCC) [39], when enhancing ANFC of a network is concerned. In near-optimal LL addition, an LL is added between a node pair such that overall APL value of the network is minimized. Similarly, in MaxBC and MaxCC, an LL is added to maximize the betweenness centrality and the closeness centrality measures [23] of the network, respectively. In the following, the LL addition strategies are studied, after the addition of 10 LLs, in the context of various arbitrary network topologies and real-world networks. Note that in all simulation experiments, ACES considers only 1-hop neighbor information. That is, $r = 1$ for node i when $NE_{i(r)}$ is concerned (see Algorithm 7.2).

7.5.1 Arbitrary Networks

We consider regular (a string and a grid networks) as well as random network topologies to study various deterministic LL addition strategies to enhance ANFC values. In the experiment, each of the networks are with 100 nodes where the flow capacity value of each link is randomly assigned in the range $[1,\ 13]$. Note that the 100-node random network is generated with link creation probability $p = 0.02$ and then flow capacity of each link is randomly assigned in the range $[1,\ 13]$. The results for all arbitrary networks are also averaged over 10 simulations.

ACES Performance: LLs with Fixed Flow Capacity

To account for ANFC improvement after addition of 10 deterministic LLs in various arbitrary network topologies such as string, grid, and random networks, we deploy MaxCap, ACES, near-optimal, MaxBC, and MaxCC based LL addition strategies. Note that the LLs are added with the fixed flow capacities of 10 units ($\mathcal{C}_{Fix} = 10$).

Figure 7.5 depicts the percentage improvement of the ANFC values after the addition of 10 LLs, and each with a fix flow capacity of 10 units, with various deterministic LL addition strategies. It can be seen that the improvement in the ANFC values is the

[5]Hereafter, the *near-optimal decision based LL addition strategy to minimize the network APL* is mentioned as the *near-optimal decision* or the *near-optimal* throughout the chapter.

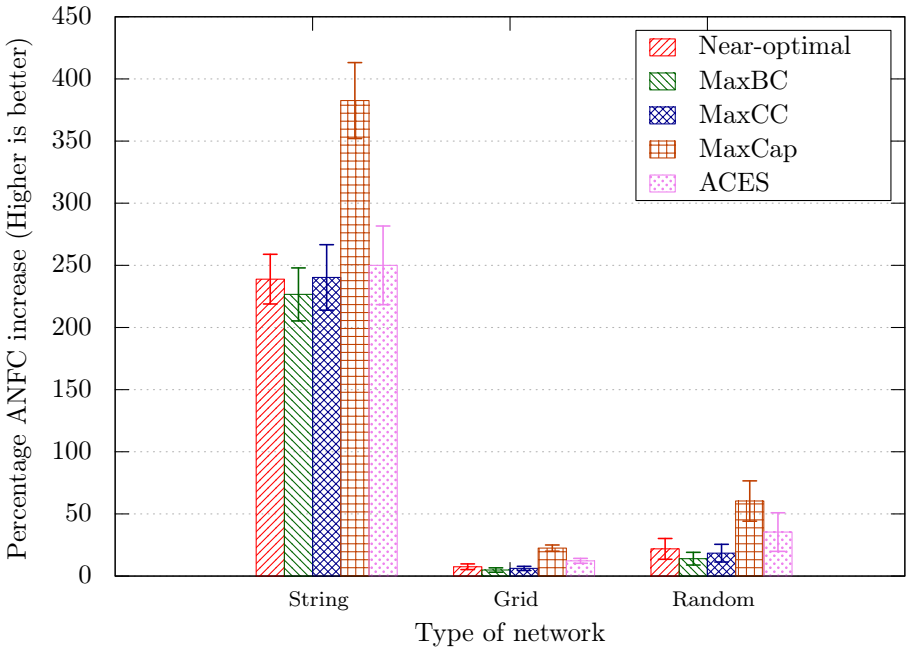


Figure 7.5: ANFC performance after addition of 10 LLs with various deterministic strategies in the context of three arbitrary network topologies: a string network, a grid network, and a random network. Note that each LL is added with fixed value of flow capacity (Flow capacity value is 10 for each LL).

highest when MaxCap based LL addition is concerned. ACES based LL addition, on the other hand, also enhances the ANFC values. However, the percentage deviation from the MaxCap based LL addition is more when string topology network is concerned. Other deterministic LL addition strategies fail to improve the ANFC values in the arbitrary networks.

ACES Performance: LLs with Variable Flow Capacity

Similarly, Figure 7.6 shows ANFC values after the addition of 10 LLs with five deterministic LL addition strategies. Note that the assigned flow capacity for each added LL, in Figure 7.6, is equal to the maximum of the flow capacity values of all links attached to either of the LL connecting nodes (i.e., $\mathcal{C}_{Var}$).

It can be seen from the figure that the percentage improvement in the ANFC value is the highest when MaxCap is concerned. ACES also improves the ANFC values. By contrast, the deviation from MaxCap is more in a string topology network. Other LL addition strategies fail to improve the ANFC values for the other three arbitrary networks.

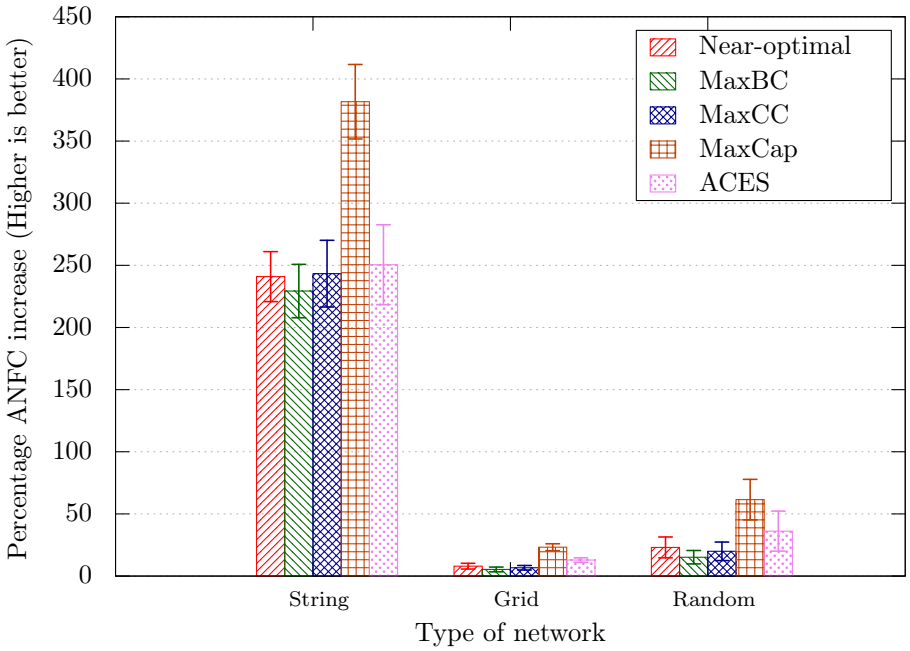


Figure 7.6: ANFC performance after addition of 10 LLs with various deterministic strategies in the context of three arbitrary network topologies. The flow capacity of each LL is equal to the highest flow capacity of links attached to the either of the LL connecting nodes.

Performance on Average Path Length Reduction

APL of a network is evaluated as the EHD between a node pair averaged over the network. Lower the value of APL, delay in transmission/transport can be lowered. By adding a few LLs in a network, APL of the network can be greatly reduced.

Figure 7.7 shows performance of all deterministic LL addition strategies when percentage reduction in APL values are concerned. Note that as MaxCap and ACES take into account fixed or variable flow capacity while deciding the location of an LL, corresponding strategies are presented in Figure 7.7 as $MaxCap_{Fix}$, $MaxCap_{Var}$, $ACES_{Fix}$, and $ACES_{Var}$, respectively.

It can be seen from Figure 7.7 that the near-optimal strategy performs the best in minimizing the network APL. The MaxBC and MaxCC based deterministic LL addition strategies also reduce the network APL comparable with the near-optimal strategy. By contrast, as MaxCap and ACES are mostly concerned with improving the ANFC value for a network, they fail to significantly reduce the network APL.

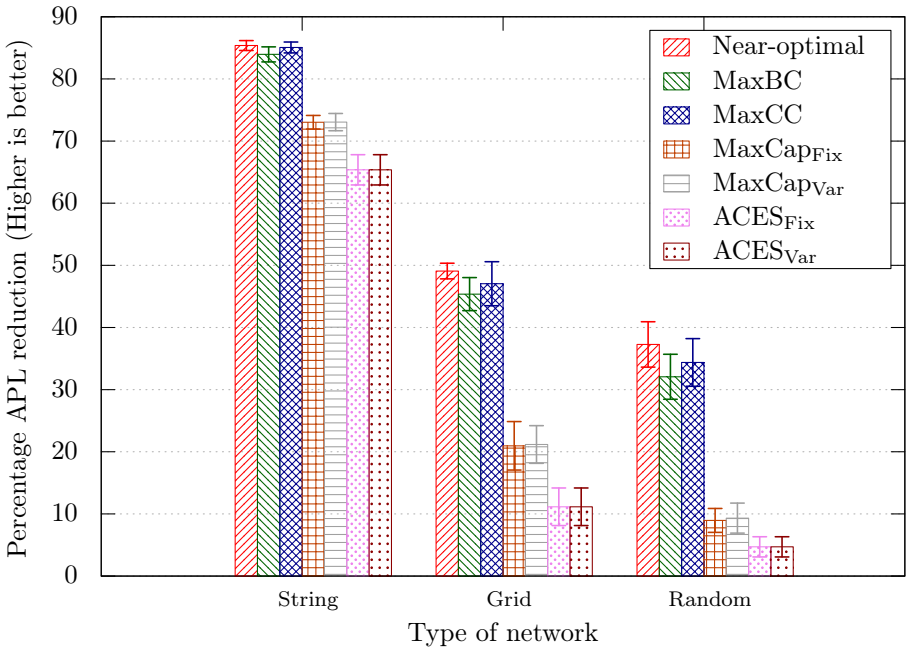


Figure 7.7: APL performance in the context of three arbitrary network topologies where 10 LLs are added with fixed or variable flow capacities.

In the following, performance of ACES is evaluated in six real-world road networks. Further, a comparison study is also carried out with the different r-hop neighbors when ACES is concerned.

7.5.2 Real-world Road Networks

We also study all LL addition strategies (i.e., MaxCap, ACES, near-optimal, MaxBC, and MaxCC), with 10 LLs, in the context of six real-world road networks (Bangalore, Delhi, London, Paris, New York, and Manhattan) as tabulated in Table 7.2.

All road networks, as listed in Table 7.2, are exported from OpenStreetMap [59] and the required information is then extracted to create the corresponding network graphs. A junction, in this context, represents a node, and a road connecting two junctions corresponds to a link. The flow capacity of each link is set based on the corresponding road-type. That is, a wider road in a road network is assigned higher flow capacity value and vice versa. Table 7.3 lists all possible road-types and associated flow capacity values. Note that the types of roads are listed in the descending order of their width.

Table 7.2
Characteristic features of six real-world road networks

Name of the City	Number of Nodes	Number of Links
Bangalore	103	139
Delhi	101	120
London	99	127
Paris	103	119
New York	100	136
Manhattan	104	152

Table 7.3
Various road-types and corresponding flow capacity values

Road-types	Flow Capacity
Motor-ways	13
Motor-way Links	12
Trunk	11
Primary	10
Secondary/Link Roads	9
Tertiary/Tertiary Links	8
Unclassified	7
Residential/Business	6
Service	5
Living Street	4
Pedestrian	3
Cycle-way	2
Foot-way/Track/Steps/Paths	1

In the LL addition experiment, the flow capacity of an LL can be assigned based on the strategy as mentioned in Section 7.3. Hence, in case of fixed flow capacity assignment to an LL (i.e., $\mathcal{C}_{Fix}$), the assigned flow capacity value is 10 which represents the *primary road-type*. The selection of fixed flow capacity value is based on the observations that most roads in real-world road networks are of the type primary roads [59]. On the other hand, in case of variable flow capacity assignment to an LL (i.e., $\mathcal{C}_{Var}$), the LL is assigned a flow capacity value equal to the widest road that is linked to either of the junction nodes where an LL is to be connected.

ACES Performance: LLs with Fixed Flow Capacity

When LLs are added with fixed flow capacity in various road networks, it can be seen that MaxCap enhances ANFC value to the maximum. We add 10 LLs, in each road network, with various LL addition strategies such as MaxCap, ACES, near-optimal, MaxBC, and MaxCC. Figure 7.8 shows ANFC improvement results for different road networks with 10 LLs, each with fixed flow capacity value. In the simulation, $\mathcal{C}_{Fix}$ value is assumed to be 10.

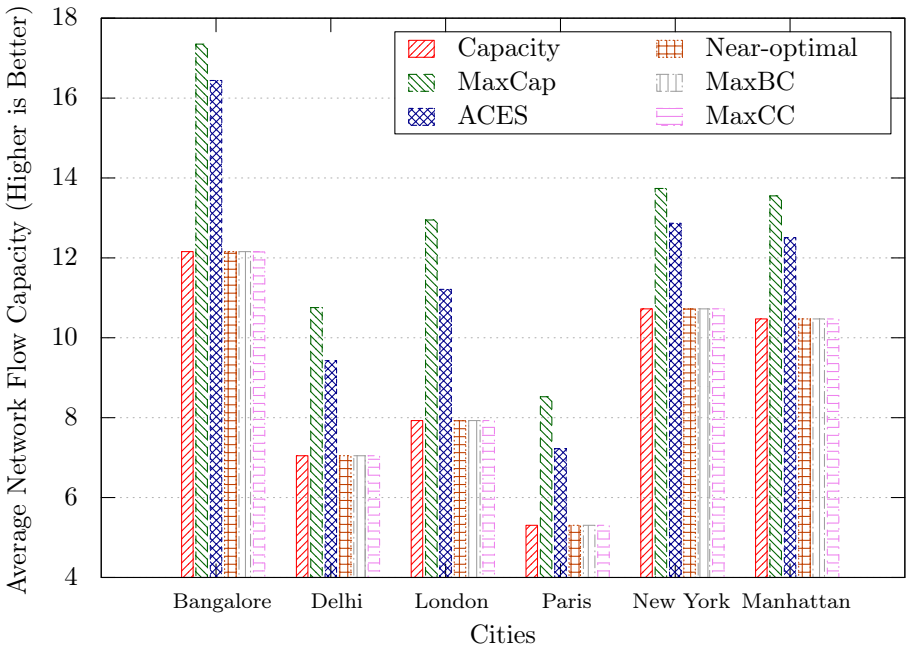


Figure 7.8: ANFC performance of various LL addition strategies in the context of six real-world road networks. Note that 10 LLs are added with fixed value of flow capacity (Flow capacity value is 10 for each LL).

From Figure 7.8, it can be observed that MaxCap strategy improves ANFC value by nearly 42.73% (Bangalore) with respect to the base network without any LLs (see the first histogram, in each road network, with the legend *Capacity*). For all other road networks, MaxCap improves over corresponding base networks approximately 52.65% (Delhi), 63.34% (London), 60.67% (Paris), 28.09% (New York), and 29.40% (Manhattan). On the other hand, ACES based LL addition deviates from MaxCap with only 5.24% (Bangalore), 12.36% (Delhi), 9.71% (London), 13.13% (Paris), 3.00% (New York), and 5.91% (Manhattan). However, as can be seen from the figure, other LL addition strategies fail to achieve noticeable improvement in ANFC.

Performance of LL Length-Type Product (LTP): Figure 7.9 shows results on the product of LL-length and LL road-type (i.e., flow capacity) for each road network. LTP can be considered analogous to the bandwidth-delay product (BDP) [67] in a communication network.

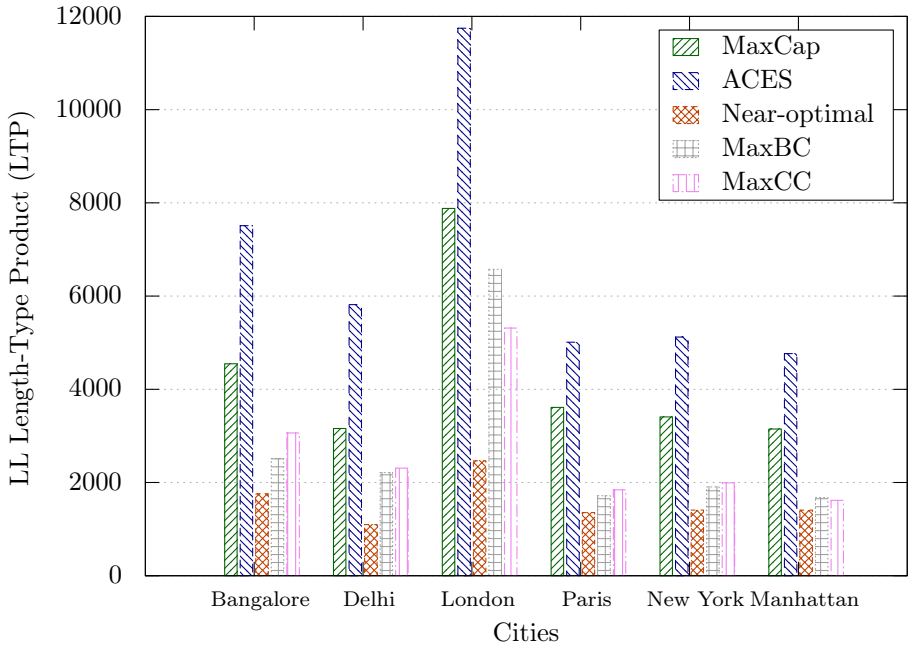


Figure 7.9: LTP performance of various LL addition strategies (with fixed flow capacity) in the context of six real-world road networks.

BDP represents how much data can be successfully transferred over a communication link. Therefore, in order to get high data rate or throughput, BDP should be sufficiently large. In case of road networks, bandwidth can be the road-type of a link and delay can be idealized by the length of the link. Hence, to achieve higher ANFC, LTP should have a larger value.

In Figure 7.9, we show the LTP performance for all LL addition strategies in six real-world road networks. From the figure, it can be observed that ACES outperforms other LL addition strategies considered in this chapter. In particular, ACES gains approximately 65.12% (Bangalore), 84.16% (Delhi), 49.05% (London), 38.68% (Paris), 50.30% (New York), and 51.38% (Manhattan) with respect to the MaxCap after adding 10 LLs with fixed value of flow capacity. Hence, the ACES strategy performs better when LTP is concerned.

ACES Performance: LLs with Variable Flow Capacity

Next, 10 LLs with variable flow capacity values are added to the road networks with the five LL addition strategies. Figure 7.10 depicts the performance of all strategies when ANFC is concerned. It can be noticed that MaxCap and ACES outperform other LL addition strategies.

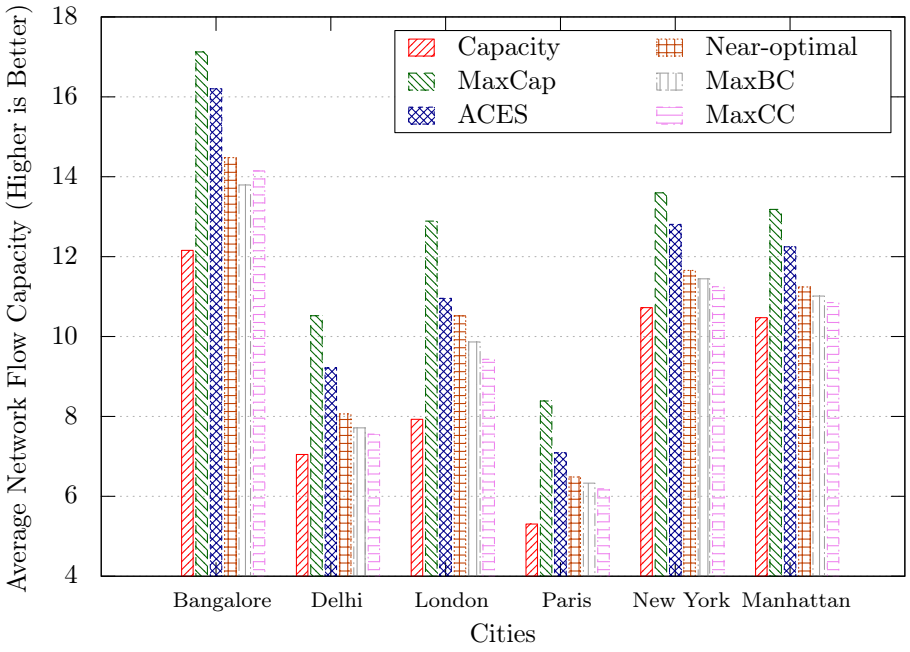


Figure 7.10: ANFC performance of various LL addition strategies in the context of six real-world road networks. The flow capacity of an LL is equal to the highest flow capacity of links attached to the either of the LL connecting nodes.

We also study the road-type distribution of LLs added with variable flow capacity values. From Figure 7.11, road-type distribution of any road network can also be identified. Note that road-type distribution of LLs with fixed flow capacity is not shown as all LL road-types are of type primary roads (Type 10).

For instance, many roads in New York are secondary/link roads (Type 9) as most of the LLs that are added are of same type (see Figure 7.11). Similarly, many roads in London can be identified as of type residential/business roads (Type 6). However, in case of Bangalore, Delhi, and Paris, LLs are distributed to different road-types. On the other hand, Manhattan city is mostly dominated by residential/business (Type 6) and tertiary links (Type 8).

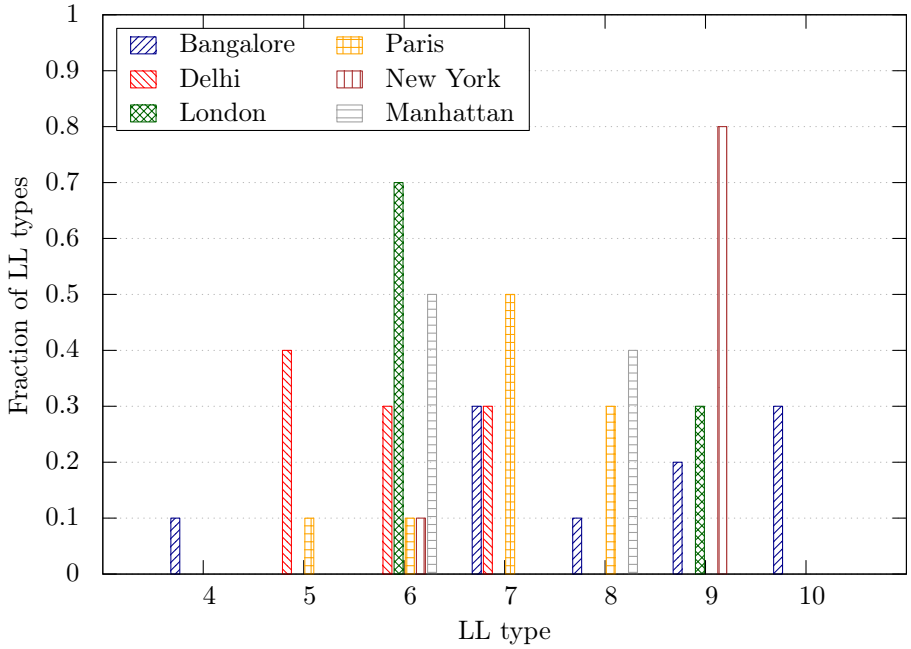


Figure 7.11: LL road-type distribution with ACES, in six real-world road networks, after addition of 10 LLs with variable flow capacity.

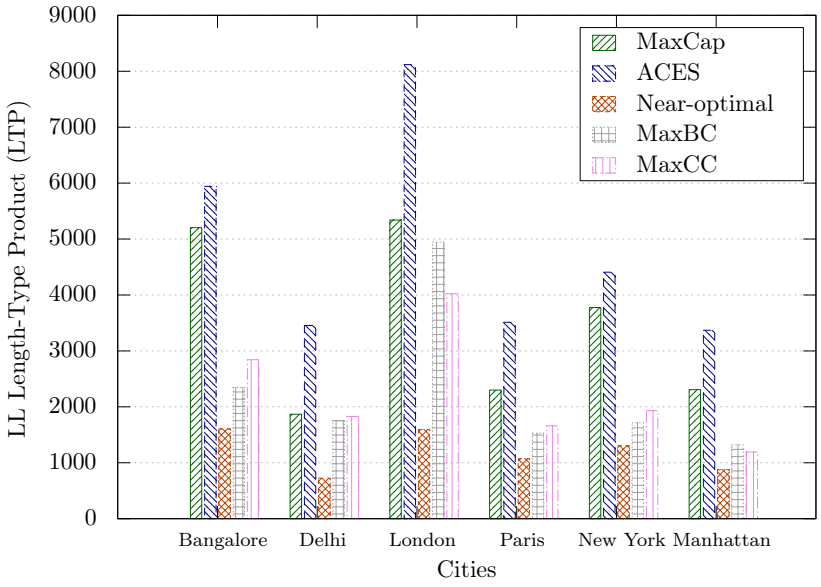


Figure 7.12: LTP performance of various LL addition strategies (with variable flow capacity) in the context of six real-world road networks.

Performance of LL Length-Type Product (LTP): Figure 7.12 shows results for LTP for different LL addition strategies with variable flow capacity for each LL added in six different road networks. From the figure, it can be noticed that ACES performs better than all other LL addition strategies. Moreover, ACES is approximately 14.19% (Ban-

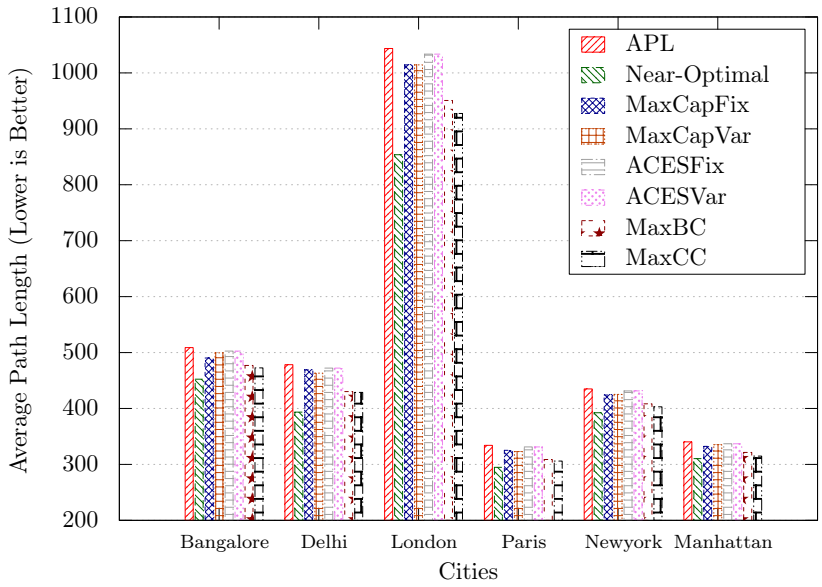


Figure 7.13: APL performance in the context of six real-world road networks where 10 LLs are added with fixed or variable flow capacities.

galore), 84.89% (Delhi), 52.06% (London), 52.67% (Paris), 16.73% (New York), and 46.01% (Manhattan) better with respect to the MaxCap strategy.

Performance on Average Path Length Reduction

In the simulation experiment, 10 LLs are added based on various LL addition strategies. Note that APL results do not consider road-type of a link. However, as MaxCap and ACES take into account road-types while deciding the location of an LL, we represent corresponding strategies in Figure 7.13 as $MaxCap_{Fix}$, $MaxCap_{Var}$, $ACES_{Fix}$, and $ACES_{Var}$, respectively.

From Figure 7.13, it can be seen that near-optimal outperforms all other LL addition strategies when minimizing APL of a network is concerned. In particular, near-optimal decision based LL addition reduces APL values approximately 11.10% (Bangalore), 17.73% (Delhi), 18.20% (London), 11.79% (Paris), 9.80% (New York), and 8.80% (Manhattan) with respect to the APL values of the networks without any LLs (see the first histogram, in each road network, with the legend APL). However, MaxCap and ACES do not achieve a significant reduction in APL as compared to near-optimal (Maximum deviation is approximately 20%).

Table 7.4
ACES performance with r-hop neighbors: LL with fixed flow capacity

City	**ANFC values for MaxCap based LL Addition**	**ANFC values for ACES based LL Addition with r-hop Neighbors**								**% Deviation w.r.t. MaxCap**
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	
Bangalore	17.3507	**16.4411**	16.2816	16.3992	16.4182	16.4182	16.4182	16.4182	16.4182	5.24
Delhi	10.7566	**9.4267**	9.2764	9.2804	9.2879	9.2879	9.2879	9.2879	9.2879	12.36
London	12.9481	11.2127	11.3119	11.2993	11.4646	11.4616	11.4783	11.5304	**11.6910**	9.71
Paris	8.5248	7.2262	7.1808	7.1808	7.1808	7.1808	**7.4005**	7.4005	7.4005	13.19
New York	13.7368	12.8659	13.1766	13.2588	**13.3244**	13.1180	13.1596	13.1339	13.1339	3.00
Manhattan	13.5513	12.5090	12.7366	12.6878	12.6979	**12.7504**	12.7504	12.7504	12.7504	5.91

Table 7.5
ACES performance with r-hop neighbors: LL with variable flow capacity

City	**ANFC values for MaxCap based LL Addition**	**ANFC values for ACES based LL Addition with r-hop Neighbors**								**% Deviation w.r.t. MaxCap**
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	
Bangalore	17.1234	16.2018	15.9832	16.1917	**16.2105**	16.2105	16.2105	16.2105	16.2105	5.33
Delhi	10.5214	**9.2156**	9.0467	9.0125	9.0477	9.0477	9.0477	9.0477	9.0477	12.41
London	12.8887	10.9563	11.0293	11.0280	11.1773	11.1629	11.1989	11.3061	**11.4123**	11.45
Paris	8.3853	7.0889	7.0701	7.0701	7.0701	7.0701	**7.2861**	7.2861	7.2861	13.11
New York	13.5980	12.8085	13.0830	13.1640	**13.2303**	13.0463	13.1010	13.1008	13.1008	2.70
Manhattan	13.1830	12.2446	12.4302	12.4356	12.4576	**12.4804**	12.4804	12.4804	12.4804	5.33

7.6 Observations and Discussion

It can be observed from Section 7.5 that ACES significantly enhances ANFC and LTP in the context of various real-world road networks. In particular, ACES performance is also comparable with respect to MaxCap based deterministic LL addition strategy when ANFC is concerned (Maximum percentage deviation is 15%). On the other hand, rest of the deterministic LL addition strategies such as near-optimal, MaxBC, and MaxCC, cannot guarantee improvement in the flow capacity of a network.

We also analyze ACES performance with different r-hop neighbors in the context of fixed as well as variable flow capacity assignment to an LL. Tables 7.4 and 7.5 list ANFC values for different r-hop neighbor information. In the simulation experiment, we consider up to 8-hop neighbors of a pendant node. The percentage deviations of ACES based highest ANFC values (shown as boldface in Tables 7.4 and 7.5) are taken with respect to the MaxCap based ANFC values. It can be observed, from both tables, that various road networks return highest ANFC values with different r-hop information.

Moreover, the time complexity of running ACES is $\mathcal{O}(k \times N^2)$ in an N-node network where k LLs are added. However, MaxCap maximizes ANFC value of a network with a cost of $\mathcal{O}(k \times N^5)$ time to add k LLs. Therefore, MaxCap is less efficient as compared to ACES when real-world deployment of LLs takes place for enhancing ANFC in a limited time frame. Other deterministic LL addition strategies such as near-optimal, MaxBC, and MaxCC are also significantly time complex, in the order of $\mathcal{O}(k \times N^4 \log N)$ [39], and fail to match with the ANFC performance when MaxCap and ACES based LL addition strategies are concerned.

Furthermore, ACES performance on LTP also outperforms other LL addition strategies. As mentioned in Section 7.5.2, LTP resembles BDP of a network which in turn identifies flow capacity of a link. From Figures 7.9 and 7.12, we can see that ACES based LL addition strategy is more capacitative with respect to other LL addition strategies. Therefore, ACES can be used to construct efficient high capacity flyovers when real-world road networks are concerned. ACES can also find applications in transportation networks, communication networks, and many other man-made networks.

The major concern about the real-world deployment of ACES strategy is the improvement in traffic congestion. Addition of new links such as flyovers in road networks may not enhance the ANFC value. On the contrary, according to the Braess' paradox [68], new link addition in networks sometimes results in reduced ANFC. Braess' paradox states that if each vehicle in a road network intends to use the optimal path to reach the destination, then the resultant time taken to reach the destination may not be minimum. As a result of that overall network congestion also increases. Therefore, taking the principle of Braess' paradox into consideration, the ACES strategy can be modified to achieve better ANFC performance along with better traffic distribution.

7.7 Summary

In this chapter we proposed a novel LL addition heuristic, average flow capacity enhancement using small-world characteristics (ACES), which significantly enhances the average network flow capacity (ANFC) of an N-node network with only $\mathcal{O}(N^2)$ time while adding an LL. Furthermore, ANFC value with ACES deviates maximum of 15% as compared to the maximum flow capacity (MaxCap) strategy which adds an LL deterministically in $\mathcal{O}(N^5)$ time. We also studied the LL length-type product (LTP) to understand the flow capacity of a deployed LL. We found that ACES outperforms MaxCap as well as other deterministic LL addition strategies such as near-optimal, MaxBC, and MaxCC, when enhancing LTP is concerned. ACES can be deployed in many real-world application scenarios where enhancing ANFC and LTP, in limited time, are very crucial.

Chapter 8

Conclusions and Future Research Directions

8.1 Conclusions

This thesis work investigates some of the open questions in the context of the evolution of finite sized complex networks. In particular, we observed transition of fixed sized complex networks by adding a few long-ranged links (LLs) with random as well as based on greedy decision making. We found that greedy decision making, based on certain network parameters such as average path length (APL) and average network flow capacity (ANFC), transforms a regular finite sized network to a scale-free networks. However, random decision based LL addition fails to transform a network to a scale-free network. In the following, we summarize major contributions of the thesis.

1. We found that a sequence of greedy optimal/near-optimal decision making resulted in the transformation of a small-world network to a scale-free network. We added a few LLs in a finite sized string topology network (STN), with the greedy near-optimal LL addition strategy, to achieve an APL-optimal small-world network. We observed, during APL-optimal LL addition, that most of the LLs inclined toward a particular node, at the $0.8N^{th}$ in an N-node STN, due to the long-ranged link affinity (LRA) and thus, the STN gradually transformed to an APL-optimal scale-free network by introducing hub node at the $0.8N^{th}$ location. However, random decision based LL addition failed to achieve APL-optimal small-world network.

2. We also observed that, with length constrained LL addition, a regular network evolves in the following manner: regular network $\rightarrow$ small-world network $\rightarrow$ scale-free network with truncated degree distribution $\rightarrow$ fully connected network.

3. While experimenting with the greedy near-optimal decision based LL addition on a finite sized STN, we discovered that the location of the first LL is always uniquely

find $0.2N^{th}$ and $0.8N^{th}$ nodes in an N-node STN. We called the unique nodes as *anchor nodes*. We also analytically found the fractional locations of the anchor nodes at 0.2071 and 0.7929. During the APL-optimal evolution of an STN, one of the anchor nodes get transformed to a hub node.

4. Based on the simulation and analytical observations on the creation of APL-optimal small-world networks, we proposed a heuristic strategy, sequential deterministic LL addition (SDLA), to efficiently transform an N-node STN to an APL-optimal small-world network only in $\mathcal{O}(k \times N)$ time by adding k LLs.

5. We analytically determined the locations of a single and two optimal LLs in a string topology wireless sensor network (WSN) to minimize average path length to the base station (APLB) value. Our analytical observations also match with the simulation as well as approximate results. Further, we studied the effect of power constrained LL addition in a string topology WSN where we found that a natural tradeoff exists between minimizing APLB value and total expended transmission power to construct an optimal LL. We found, with no power constraints, fixed fractional locations of a single optimal LL is at (0.33, BS), and the two concurrent LLs can be found at (0.2, BS) and (0.6, BS).

6. To the end of this thesis, we also applied small-world characteristics in order to enhance the ANFC value and the length-type product (LTP) of weighted undirected complex networks. We applied an exhaustive LL addition strategy, maximum flow capacity (MaxCap), to maximize ANFC value of a network. Further, based on the observations from MaxCap, we proposed a heuristic, average flow capacity enhancement using small-world characteristics (ACES), to efficiently enhance ANFC and LTP values.

8.2 Future Research Directions

In the following, a few possible extensions of this thesis work are discussed.

1. Most of the existing solutions, to create an LL, consider unweighted networks.

However, in real-world scenarios, LL may not always be unweighted. How to realize an LL edge-weight in a realistic small-world network, can be a possible extension of the thesis.

2. In this thesis, we studied evolution of a fixed sized regular network to a fully connected network. However, there is no existing theoretical framework to explain such network evolution. Modeling a mathematical framework to study the evolution of finite sized complex networks can be an interesting research problem.

3. Existence of anchor nodes, as in a 1-D topology such as an STN, is not known in a 2-D network. Therefore, another very compelling open question is that whether there exists any anchor node in a 2-D network, and if so, how to find the location of the anchor node.

4. SDLA algorithm was developed for an unweighted STN. Therefore, it will be interesting to extend the SDLA algorithm in a general setting where weighted arbitrary networks can be considered. Testbed implementation and performance analysis of SDLA algorithm can also be considered as a possible future extension.

5. Optimal power constrained LL addition was studied, in a string topology WSN, for a single as well as two LLs. Design of a generalized algorithm to add k LLs in such a network is an open research problem. Further, analytical derivation can also be extended to identify optimal locations of power efficient LLs in a 2-D network regime.

6. Real-world deployment of deterministic LL addition strategies such as ACES always may not improve traffic/congestion situation because of the Braess' paradox which states that addition of additional LL could actually increase overall congestion in a network. Therefore, the design of more efficient capacity-enhanced LL addition algorithm to overcome the Braess' paradox is another open research problem. A few more key metrics, such as travel time, traffic management issues, and other environmental impacts, while applying small-world characteristics in the design of efficient real-world road networks, can also be exercised.

Bibliography

[1] B. S. Manoj, A. Chakraborty, and R. Singh, *Complex networks: A networking and signal processing perspective*. Prentice Hall PTR, New Jersey, USA, 2018.

[2] S. Milgram, "The small world problem," *Psychology Today*, vol. 2, no. 1, pp. 60–67, May 1967.

[3] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-world networks," *Nature*, vol. 393, no. 6684, pp. 440–442, June 1998.

[4] M. E. J. Newman and D. J. Watts, "Renormalization group analysis of the small-world network model," *Physics Letters A*, vol. 263, no. 4, pp. 341–346, December 1999.

[5] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, October 1999.

[6] A.-L. Barabási and E. Bonabeau, "Scale-free networks," *Scientific American*, vol. 288, no. 5, pp. 60–69, May 2003.

[7] P. J. Denning, "Network laws," *Communications of the ACM*, vol. 47, no. 11, pp. 15–20, November 2004.

[8] A.-L. Barabási, "Network science," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 371, no. 1987, pp. 1–3, February 2013.

[9] F. Papadopoulos, M. Kitsak, M. Á. Serrano, M. Boguñá, and D. Krioukov, "Popularity versus similarity in growing networks," *Nature*, vol. 489, no. 7417, pp. 537–540, September 2012.

[10] A.-L. Barabási, "Luck or reason," *Nature*, vol. 489, no. 7417, pp. 507–508, September 2012.

[11] A. Chakraborty and B. S. Manoj, "The reason behind the scale-free world," *IEEE Sensors Journal*, vol. 14, no. 11, pp. 4014–4015, November 2014.

[12] A.-L. Barabási, N. Gulbahce, and J. Loscalzo, "Network medicine: A network-based approach to human disease," *Nature Reviews Genetics*, vol. 12, no. 1, pp. 56–68, January 2011.

[13] S. Redner, "How popular is your paper? An empirical study of the citation distribution," *The European Physical Journal B – Condensed Matter and Complex Systems*, vol. 4, no. 2, pp. 131–134, July 1998.

[14] ——, "Citation statistics from 110 years of physical review," *Physics Today*, vol. 58, no. 6, pp. 49–54, June 2005.

[15] K.-I. Goh, B. Kahng, and D. Kim, "Universal behavior of load distribution in scale-free networks," *Physical Review Letters*, vol. 87, no. 27, p. 278701, December 2001.

[16] A.-L. Barabási, "Scale-free networks: A decade and beyond," *Science*, vol. 325, no. 5939, pp. 412–413, July 2009.

[17] G. Caldarelli, *Scale-free networks: Complex webs in nature and technology*. Oxford University Press, 2007.

[18] R. Albert, H. Jeong, and A.-L. Barabási, "Internet: Diameter of the world-wide web," *Nature*, vol. 401, no. 6749, pp. 130–131, September 1999.

[19] C. A. Hidalgo, "Conditions for the emergence of scaling in the inter-event time of uncorrelated and seasonal systems," *Physica A: Statistical Mechanics and its Applications*, vol. 369, no. 2, pp. 877–883, September 2006.

[20] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Physical Review E*, vol. 74, no. 3, p. 036104, September 2006.

[21] ——, "A symmetrized snapshot of the structure of the internet at the level of autonomous systems, reconstructed from bgp tables posted by the university of oregon route views project," *Unpublished*, July 2006.

[22] "Gephi datasets," https://github.com/gephi/gephi/wiki/Datasets, accessed: 2017-11-09.

[23] L. C. Freeman, "Centrality in social networks conceptual clarification," *Social Networks*, vol. 1, no. 3, pp. 215–239, 1978-1979.

[24] W. Tan, M. B. Blake, I. Saleh, and S. Dustdar, "Social-network-sourced big data analytics," *IEEE Internet Computing*, vol. 17, no. 5, pp. 62–69, September 2013.

[25] J. Mei and J. M. F. Moura, "Signal processing on graphs: Causal modeling of unstructured data," *IEEE Transactions on Signal Processing*, vol. 65, no. 8, pp. 2077–2092, April 2017.

[26] M. Barthélémy and L. A. N. Amaral, "Small-world networks: Evidence for a crossover picture," *Physical Review Letters*, vol. 82, no. 15, pp. 3180–3183, April 1999.

[27] N. C. Wormald, "Models of random regular graphs," *Surveys in Combinatorics*, pp. 239–298, 1999.

[28] P. Erdös and A. Rényi, "On random graphs I," *Publicationes Mathematicae (Debrecen)*, vol. 6, pp. 290–297, 1959.

[29] ——, "On the evolution of random graphs," *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, vol. 5, pp. 17–61, 1960.

[30] X. F. Wang and G. Chen, "Complex networks: Small-world, scale-free and beyond," *IEEE Circuits and Systems Magazine*, vol. 3, no. 1, pp. 6–20, First Quarter 2003.

[31] J. Kleinberg, "The small-world phenomenon: An algorithmic perspective," in *Proceedings of the 32nd annual ACM symposium on Theory of computing*, pp. 163–170, May 2000.

[32] J. M. Kleinberg, "Navigation in a small world," *Nature*, vol. 406, no. 6798, pp. 845–845, August 2000.

[33] G. Bianconi and A.-L. Barabási, "Competition and multiscaling in evolving networks," *Europhysics Letters*, vol. 54, no. 4, pp. 436–442, May 2001.

[34] G. Caldarelli, A. Capocci, P. De Los Rios, and M. A. Munoz, "Scale-free networks from varying vertex intrinsic fitness," *Physical Review Letters*, vol. 89, no. 25, p. 258702, December 2002.

[35] G. Timár, S. N. Dorogovtsev, and J. F. F. Mendes, "Scale-free networks with exponent one," *Physical Review E*, vol. 94, no. 2, p. 022302, August 2016.

[36] A. Helmy, "Small worlds in wireless networks," *IEEE Communications Letters*, vol. 7, no. 10, pp. 490–492, October 2003.

[37] F. Comellas and M. Sampels, "Deterministic small-world networks," *Physica A: Statistical Mechanics and its Applications*, vol. 309, no. 1, pp. 231–235, June 2002.

[38] Z. Zhang, L. Rong, and C. Guo, "A deterministic small-world network created by edge iterations," *Physica A: Statistical Mechanics and its Applications*, vol. 363, no. 2, pp. 567–572, May 2006.

[39] N. Gaur, A. Chakraborty, and B. S. Manoj, "Delay optimized small-world networks," *IEEE Communications Letters*, vol. 18, no. 11, pp. 1939–1942, November 2014.

[40] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, December 1959.

[41] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT Press, USA, July 2009.

[42] A. Meyerson and B. Tagiku, "Minimizing average shortest path distances via shortcut edge addition," *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pp. 272–285, August 2009.

[43] M. Papagelis, F. Bonchi, and A. Gionis, "Suggesting ghost edges for a smaller world," in *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, pp. 2305–2308, October 2011.

[44] M. Papagelis, "Refining social graph connectivity via shortcut edge addition," *ACM Transactions on Knowledge Discovery from Data*, vol. 10, no. 2, p. 12, October 2015.

[45] A. Vinel, L. Lan, and N. Lyamin, "Vehicle-to-vehicle communication in c-acc/platooning scenarios," *IEEE Communications Magazine*, vol. 53, no. 8, pp. 192–197, August 2015.

[46] A. Chakraborty, Vineeth B. S., and B. S. Manoj, "Analytical identification of anchor nodes in a small-world network," *IEEE Communications Letters*, vol. 20, no. 6, pp. 1215–1218, June 2016.

[47] ——, "Influence of greedy reasoning on network evolution," in *Large Scale Complex Network Analysis (LSCNA2015)*. Academic Publishers, pp. 81–84, December 2015.

[48] A. Chakraborty and B. S. Manoj, "An efficient heuristics to realize near-optimal small-world networks," in *21st National Conference on Communications (NCC)*, pp. 1–5, February 2015.

[49] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey," *Ad-Hoc Networks*, vol. 7, no. 3, pp. 537–568, May 2009.

[50] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, March 2002.

[51] A. Bhattacharya, S. M. Ladwa, R. Srivastava, A. Mallya, A. Rao, D. G. R. Sahib, S. Anand, and A. Kumar, "SmartConnect: A system for the design and deployment of wireless sensor networks," in *Fifth IEEE International Conference on Communication Systems and Networks (COMSNETS)*, pp. 1–10, January 2013.

[52] R. Chitradurga and A. Helmy, "Analysis of wired short cuts in wireless sensor networks," in *Proceedings of the The IEEE/ACS International Conference on Pervasive Services*, pp. 167–176, July 2004.

[53] D. L. Guidoni, A. Boukerche, F. S. H. Souza, R. A. F. Mini, and A. A. F. Loureiro, "A small world model based on multi-interface and multi-channel to design heterogeneous wireless sensor networks," in *IEEE Global Telecommunications Conference (GLOBECOM)*, pp. 1–5, December 2010.

[54] D. L. Guidoni, R. A. F. Mini, and A. A. F. Loureiro, "On the design of resilient heterogeneous wireless sensor networks based on small world concepts," *Computer Networks*, vol. 54, no. 8, pp. 1266–1281, June 2010.

[55] D. L. Guidoni, R. A. F. Mini, and R. A. F. Mini, "Applying the small world concepts in the design of heterogeneous wireless sensor networks," *IEEE Communications Letters*, vol. 16, no. 7, pp. 953–955, July 2012.

[56] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on wireless communications*, vol. 1, no. 4, pp. 660–670, October 2002.

[57] G. Sharma and R. Mazumdar, "Hybrid sensor networks: A small world," in *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 366–377, May 2005.

[58] C. C. Enz, A. El-Hoiydi, J.-D. Decotignie, and V. Peiris, "WiseNET: an ultralow-power wireless sensor network solution," *Computer*, vol. 37, no. 8, pp. 62–70, August 2004.

[59] S. Babu and B. S. Manoj, "On the topology of indian and western road networks," in *8th IEEE International Conference on Communication Systems and Networks (COMSNETS) Intelligent Transportation Systems Workshop*, pp. 1–6, January 2016.

[60] Q. Li, Z. Zeng, and B. Yang, "Hierarchical model of road network for route planning in vehicle navigation systems," *IEEE Intelligent Transportation Systems Magazine*, vol. 1, no. 2, pp. 20–24, Summer 2009.

[61] Q. Song and X. Wang, "Efficient routing on large road networks using hierarchical communities," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 1, pp. 132–140, March 2011.

[62] S. Coogan and M. Arcak, "A compartmental model for traffic networks and its dynamical behavior," *IEEE Transactions on Automatic Control*, vol. 60, no. 10, pp. 2698–2703, October 2015.

[63] S. Wang, S. Djahel, Z. Zhang, and J. McManis, "Next road rerouting: A multiagent system for mitigating unexpected urban traffic congestion," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 10, pp. 2888–2899, October 2016.

[64] Z.-Y. Jiang and J.-F. Ma, "Deployment of check-in nodes in complex networks," *Scientific Reports*, vol. 7, p. 40428, January 2017.

[65] N. Deo, *Graph theory with applications to engineering and computer science*. PHI Learning Pvt. Ltd., 2004.

[66] J. Edmonds and R. M. Karp, "Theoretical improvements in algorithmic efficiency for network flow problems," *Journal of the ACM*, vol. 19, no. 2, pp. 248–264, April 1972.

[67] L. L. Peterson and B. S. Davie, *Computer networks: A systems approach*, 5th ed. Morgan Kaufmann Publishers Inc., 2011.

[68] R. Steinberg and W. I. Zangwill, "The prevalence of Braess' paradox," *Transportation Science*, vol. 17, no. 3, pp. 301–318, August 1983.