

**ANT BASED ALGORITHM AND ROBUSTNESS  
METRIC IN SPARE CAPACITY ALLOCATION  
FOR SURVIVABLE ROUTING**

---

A thesis submitted in partial fulfilment of the  
requirements for the Degree  
of Doctor of Philosophy in Electrical and Computer Engineering  
in the University of Canterbury  
by Zhiyong (William) Liu  
University of Canterbury

2010

---



# Table of Contents

List of Figures.....	iv
List of Tables .....	vii
Acknowledgements .....	viii
Abstract.....	ix
Glossary .....	xi
Chapter 1 Introduction.....	1
1.1 Background.....	2
1.2 Motivation .....	3
1.3 Contributions .....	4
1.4 Thesis Outline.....	5
1.5 Published Papers.....	7
Chapter 2 Spare Capacity Allocation for Survivable Routing.....	9
2.1 Background on Network Resiliency.....	9
2.1.1 Restoration and Protection.....	10
2.1.2 Centralized vs. Distributed Protection.....	10
2.1.3 Link vs. Path Protection.....	11
2.1.4 Protection Techniques .....	12
2.1.5 Shared Risk Link Groups.....	13
2.1.6 P-cycle .....	14
2.1.7 Shared Backup Path Protection .....	15
2.2 Spare Capacity Allocation Problem.....	16
2.3 ILP models for SCA .....	18
2.3.1 Link Protection .....	18
2.3.2 Shared Backup Path Protection .....	23
2.3.3 P-Cycles.....	26
2.4 Feasible Routes Enumeration in ILP models.....	28
2.5 Capacity Sharing Information in SCA.....	29
2.6 Summary.....	30
Chapter 3 Distributed Resilience Matrix .....	33
3.1 Information on Capacity Sharing.....	33

3.1.1 Background on Capacity Sharing .....	34
3.1.2 The SPM structure .....	37
3.2 Proposed DRM structure.....	40
3.3 Extension of DRM for Multiple failures.....	48
3.4 Summary .....	55
Chapter 4 FoF-R Ant-based Routing Algorithm.....	57
4.1 Introduction on Ant Colony Optimization .....	58
4.1.1 Ant Colony Optimization.....	60
4.1.2 ACO Algorithms for Network Routing.....	61
4.2 FoF-R Ant-based Survivable Routing Algorithm .....	65
4.2.1 FoF-R Ant Agent .....	65
4.2.2 Capacity Headroom-dependent Function.....	66
4.2.3 FoF-R Ant Routing Table Structure.....	67
4.2.4 FoF-R Ant Routing Table Updating .....	69
4.2.5 FoF-R Routing Procedure .....	71
4.3 Implementation of FoF-R on OMNeT++.....	74
4.3.1 Introduction on OMNeT++.....	74
4.3.2 FoF-R Ant Models .....	74
4.3.3 FoF-R Ant Routing Parameter Setting.....	75
4.4 Simulation Results .....	76
4.5 Summary .....	85
Chapter 5 Utility of Algebraic Connectivity Metric in SCA .....	87
5.1 Topological Measures of Network Robustness.....	88
5.1.1 Structural Measures.....	88
5.1.2 Spectral Measures .....	89
5.2 The Algebraic Connectivity Metric $a(G)$ .....	90
5.2.1 The Derivation of Algebraic Connectivity.....	90
5.2.2 The Relations between $a(G)$ , $e(G)$ and $v(G)$ .....	92
5.2.3 Graph Partitioning.....	93
5.3 Properties of Algebraic Connectivity.....	94
5.4 The Mean Distance .....	95
5.5 Experiment Results .....	96

5.5.1 Successive Deletion of Nodes .....	96
5.5.2 Successive Deletion of Links.....	102
5.5.3 Repositioning of Links .....	106
5.5.4 The Combined Metric $\frac{\rho}{\lambda_2}$ .....	111
5.6 Summary.....	115
Chapter 6 Conclusion and Future work.....	117
6.1 Contributions .....	117
6.2 Future work.....	118
References .....	119

# List of Figures

Figure 1.1 Thesis outline.....	5
Figure 2.1 Link protection vs. Path protection.....	11
Figure 2.2 An example of p-cycle protection.....	14
Figure 3.1 An example of capacity sharing in a simple network with 6 nodes .....	34
Figure 3.2 An illustration of capacity categories in link $i$ .....	35
Figure 3.3 An example network of 6 nodes and 10 links.....	37
Figure 3.4 An example of SPM structure: matrices $P$ and $Q$ .....	38
Figure 3.5 An example of Spare Provision Matrix (SPM) $G$ and column vector $s$ .....	39
Figure 3.6 Example of complete $T$ .....	40
Figure 3.7 The necessary information in $T$ for links 1-3 at node “a” .....	41
Figure 3.8 Local $T_{Link_l}$ , $l=1, 2$ and $3$ in node a .....	42
Figure 3.9 Local $T_{Link_l}$ , $l=1, 4$ and $5$ in node b .....	43
Figure 3.10 Local $T_{Link_l}$ , $l=2, 6$ , and $7$ in node c .....	43
Figure 3.11 Local $T_{Link_l}$ , $l= 3,4,6,8$ and $9$ in node d .....	44
Figure 3.12 Local $T_{Link_l}$ , $l=5, 8$ , and $10$ in node e .....	44
Figure 3.13 Local $T_{Link_l}$ , $l=7, 9$ , and $10$ in node f.....	45
Figure 3.14 The WP and candidate PPs for the new request $r_{new}$ .....	46
Figure 3.15 Updated $T_{Link_l}$ for each link according to the new request $r_{new}$ .....	47
Figure 3.16 Example of working paths .....	49
Figure 3.17 Example of protection paths .....	49
Figure 3.18 Matrices of $P$ , $Q$ and $D$ .....	49
Figure 3.19 Failure scenario matrix $F$ .....	50
Figure 3.20 Spare Provision Matrix $G$ and column vector $s$ .....	50
Figure 3.21 Example of complete $T$ in DRM structure.....	51
Figure 3.22 The necessary information in $T_{Node_a}$ for links 1-3.....	51
Figure 3.23 Mapping between $T_{Node_a}$ to $T_{link_l}$ .....	51
Figure 3.24 Local $T_{link_l}$ , $l=1,2,\dots,10$ .....	52
Figure 3.25 The WP and candidate PPs for a new request $r_{new}$ .....	53
Figure 3.26 Updated $T_{link_l}$ for each link according to a new request $r_{new}$ .....	54
Figure 4.1 Finding the shortest path around an obstacle.....	58

Figure 4.2 Probability matrix at time $t$ from the obstacle scenario.....	62
Figure 4.3 An example of $R_i(h_i)$ .....	67
Figure 4.4 An example of routing table at node “a” in a simple network with 6 nodes.....	68
Figure 4.5 FoF-R ant’s routing table updating .....	69
Figure 4.6 FoF-R ant-based routing procedure.....	71
Figure 4.7 The FoF-R ant node structure.....	74
Figure 4.8 COST266 topology scenarios.....	77
Figure 4.9 JANOS-US-CA network.....	78
Figure 4.10 Total capacity usage comparison for COST266 networks .....	80
Figure 4.11 Algorithm calculation time for COST266 networks .....	81
Figure 4.12 Total capacity usage comparison for JANOS-US-CA networks .....	82
Figure 4.13 Algorithm calculation time for JANOS-US-CA network .....	82
Figure 4.14 10 reference networks .....	83
Figure 4.15 Total capacity usage vs. average nodal degree for 10 topologies .....	84
Figure 5.1 COST266 Network.....	96
Figure 5.2 JANOS-US-CA Network .....	97
Figure 5.3 Total capacity vs. average nodal degree after deleting specific nodes in COST266 reference network .....	100
Figure 5.4 Total capacity vs algebraic connectivity after deleting specific nodes in COST266 reference network .....	100
Figure 5.5 Total capacity vs. average nodal degree after deleting specific nodes in JANOS-US-CA reference network .....	101
Figure 5.6 Total capacity vs. algebraic connectivity after deleting specific nodes in JANOS-US-CA reference network .....	101
Figure 5.7 Total capacity vs algebraic connectivity after deleting specific links in COST266 reference network .....	105
Figure 5.8 Total capacity vs algebraic connectivity after deleting specific links in JANOS-US-CA reference network .....	105
Figure 5.9 COST266-1 .....	106
Figure 5.10 COST266-2 .....	107
Figure 5.11 COST266-3 .....	107
Figure 5.12 COST266-4 .....	108
Figure 5.13 COST266-5 .....	108

Figure 5.14 COST266-6.....	109
Figure 5.15 COST266-7.....	109
Figure 5.16 Total capacity vs algebraic connectivity for links repositions in COST266 reference network.....	110
Figure 5.17 20 referenced network topologies.....	111
Figure 5.18 Total capacity vs. average nodal degree.....	113
Figure 5.19 Total capacity vs. Exact mean distance/algebraic connectivity.....	114
Figure 5.20 Total capacity vs. upper bound mean distance/algebraic connectivity.....	114

# List of Tables

Table 4.1 Total capacity of four protection schemes in COST266 networks .....	79
Table 4.2 Total capacity of four protection schemes in JANOS-US-CA networks .....	81
Table 4.3 Total capacity usage comparison for 10 topologies .....	84
Table 5.1 Total capacity, algebraic connectivity and average nodal degree after deleting specific nodes in COST266 reference network .....	98
Table 5.2 Total capacity, algebraic connectivity and average nodal degree after deleting specific nodes in JANOS-US-CA reference network.....	99
Table 5.3 Total capacity, algebraic connectivity and average nodal degree after deleting specific links in COST266 reference network.....	103
Table 5.4 Total capacity, algebraic connectivity and average nodal degree after deleting specific links in JANOS-US-CA reference network .....	104
Table 5.5 Total capacity vs. algebraic connectivity for links repositions in COST266 reference network .....	110
Table 5.6 Network Information for 20 referenced networks .....	112
Table 5.7 Total capacity vs. algebraic connectivity, exact mean distance and upper bound of mean distance for 20 reference networks .....	113

# Acknowledgements

I would like to express my deep and sincere gratitude to my supervisor, Professor Harsha Sirisena, of the Department of Electrical and Computer Engineering, University of Canterbury. His wide knowledge and logical way of thinking have been of great value for me. His understanding, encouragement and personal guidance have provided a good basis for the present dissertation. I am also deeply grateful to my co-supervisor, Professor Krzysztof Pawlikowski, of the Department of Computer Science, University of Canterbury, for his great support whenever it was needed. Sincere thanks also go out to Dr Allan McInnes for his readiness to provide assistance.

I would also like to thank Professor Franco Davoli, at University of Genoa, Italy. During my three-month research at the Department of Communications, Computer and Systems Science (DIST), he has given me good advice on tackling capacity allocation problems for survivable routing.

During this dissertation work, I have collaborated with many colleagues for whom I have great regard, and I wish to extend my warmest thanks to all those who have inspired me in my work in our Network Research Group (NRG), at the University of Canterbury. This includes Tim Hong, Adam Chang, Sayan Ray and Huan Zhang.

I owe my loving thanks to my parents. Without their encouragement and understanding it would have been impossible for me to finish this work.

The financial support of the University of Canterbury is gratefully acknowledged. I also wish to thank Harmonic for the support provided through the NGNs project, the KAREN Capability Build Fund from REANNZ and PlanetLab NZ project, BuildIT and the Royal Society of New Zealand for their travel grants.

# Abstract

Network resiliency pertains to the vulnerability of telecommunication networks in the case of failures and malicious attacks. With the increasing capacity catering of network for the booming multi-services in Next Generation Networks (NGNs), reducing recovery time and improving capacity efficiency while providing high quality and resiliency of services has become increasingly important for the future network development. Providing network resiliency means to rapidly and accurately reroute the traffic via diversely routed spare capacity in the network when a failure takes down links or nodes in the working path. Planning and optimization for NGNs require an efficient algorithm for spare capacity allocation (SCA) that assures restorability with a minimum of total capacity. This dissertation aims to understand and advance the state of knowledge on spare capacity allocation in network resiliency for telecommunication core networks.

Optimal network resiliency design for restorability requires considering: network topology, working and protection paths routing and spare capacity allocation. Restorable networks should be highly efficient in terms of total capacity required for restorability and be able to support any target level of restorability. The SCA strategy is to decide how much spare capacity should be reserved on links and to pre-plan protection paths to protect traffic from a set of failures. This optimal capacity allocation problem for survivable routing is known as NP-complete. To expose the problem structure, we propose a model of the SCA problem using a matrix-based framework, named Distributed Resilience Matrix (DRM) to identify the dependencies between the working and protection capacities associated with each pair of links and also to capture the local capacity usage information in a distributed control environment. In addition, we introduce a novel ant-based heuristic algorithm, called Friend-or-Foe Resilient (FoF-R) ant-based routing algorithm to find the optimal protection cycle (i.e., two node-disjoint paths between a source-destination node pair) and explore the sharing ability among protection paths using a capacity headroom-dependent attraction and repulsion function. Simulation results based on the OMNeT++ and AMPL/CPLEX tools show that the FoF-R scheme with the DRM structure is a promising approach to solving the SCA problem for survivable routing and it gives a good trade off between solution optimality and computation speed.

Furthermore, for the SCA studies of survivable networks, it is also important to be able to differentiate between network topologies by means of a robust numerical measure that indicates the level of immunity of these topologies to failures of their nodes and links. Ideally, such a measure should be sensitive to the existence of nodes or links, which are more important than others, for example, if their failure causes the network's disintegration. Another contribution in this dissertation is to introduce an algebraic connectivity metric, adopted from the spectral graph theory, namely the 2<sup>nd</sup> smallest eigenvalue of the Laplacian matrix of the network topology, instead of the average nodal degree, to characterize network robustness in studies of the SCA problem. Extensive simulation studies confirm that this metric is a more informative parameter than the average nodal degree for characterizing network topologies in network resiliency studies.

# Glossary

ACO: Ant colony optimization

APS: automatic protection switching

BLDM: Backup Load Distribution Matrix

CR-LDP: Constraint based Label Distribution Protocol

CHF: Capacity Headroom-dependent Function

DRM: Distributed Resilience Matrix

DP: Dedicated Protection

FMT: Fault Management Table

FOF-R: Friend-or-Foe Resilient

GMPLS: Generalized Multi-Protocol Label Switching

ILP: Integer Linear Programming

MACO: Multiple Ants Colony Optimization System

MPLS: Multi-Protocol Label Switching

OSPF-TE: Open Shortest Path First- Traffic Engineering

QoS: Quality of Service

RAFT: Resource Aggregation Fault Tolerant

SBPP: Shared Backup Path Protection

SCA: Spare Capacity Allocation

SCI: Sharing with Complete Information (SCI)

SRI: Sharing with Reduced Information

SRLGs: Shared Risk Link Groups

SPM: Spare Provision Matrix

SVD: Singular Value Decomposition

WDM: Wavelength Division Multiplexing



# Chapter 1

## Introduction

Due to the increasing social dependency on telecommunication network facilities in the ongoing “information era”, network resiliency has become a very critical research area. Network resiliency gauges the ability of the network to support the committed Quality of Services (QoS) to customers continuously, even in the presence of various failures. Network resilience mechanisms include a set of methods to pre-plan and utilize the network’s spare resources to guarantee seamless communications upon failure. Failures include fiber cuts, because of construction work, natural disasters, human errors or hardware and software breakdown as described e.g., in [1-3]. Most of these failures are hard to forecast and eliminate, but it is possible to mitigate the impact of a set of specific failure scenarios by introducing network resilience techniques into the network design phase.

Traditional network resilience mechanism includes two main aspects: survivable routing and spare capacity allocation (SCA). These two phases are complementary to each other and cooperate to achieve seamless communication service operation upon failure [4]. Survivable routing refers to the incorporation of a diverse routing mechanism into the network design phase, so as to reroute the traffic upon a set of given failure scenarios. The SCA is a major component in dimensioning a survivable network when the network topology is given. It ensures enough spare capacity for the physical network or the virtual network to recover from a failure via traffic rerouting. An optimal SCA design procedure requires the efficient capacity allocation both on working and protection paths. It ensures that the working paths are routed over relatively short routes as well as efficiently restorable by the protection paths. It targets the achievement of maximal sharing among the redundant resources in the network.

This dissertation focuses on the SCA problem for survivable routing. An in-depth description of the SCA is provided and its Integer Linear Programming (ILP) model is presented. We focus on tackling the SCA problem by an ant-based heuristic routing algorithm and also on investigating a topological metric to quantify the network robustness characteristic. Our novel solutions such as the Distributed Resilience Matrix (DRM), Friend-or-Foe Resilient (FoF-R) ant-based routing algorithm and the utility of an algebraic connectivity metric in SCA have been developed and verified by extensive simulation results. We believe they are three valuable contributions to the research area of network resiliency.

## 1.1 Background

Modern telecommunication networks are equipped with ultra-high speed switches to meet the dynamically changing traffic demands. In backbone networks, connection requests are launched dynamically from the upper layers, for which routing algorithms are used to derive the corresponding capacity guaranteed tunnels that meet all the QoS requirements [5]. Offering a reliable communication platform with strong traffic engineering mechanisms and QoS guarantee leads the carrier and its customers to a revenue-generating environment. The long-term telecommunication survivors will be those who can find the balance between technological innovation, improved customer services, and efficient allocation of network resources. Therefore, network resiliency is becoming one of the most important QoS issues, and shared path protection scheme is recognized as one of the best strategies to equip the network with service resiliency by pre-planning path protection [6-9]. With a disjoint working-protection path pair, once the working path fails unexpectedly, its working traffic can be switched to the protection path and thus the corresponding services can be restored.

In recent years, we have seen great progress in network resilience technologies that enable telecommunication systems to continue operating without disruption, despite the inevitable failure events. Key network resilience protocols have been studied for at least the last two decades, and every year, the world's telecommunications journals and conferences are filled with new efforts in this area. The research results presented in this dissertation aim to contribute to that body of knowledge by offering new network resilience techniques and solutions to solve the SCA problem for survivable routing. In addition, a more informative

topological metric i.e., algebraic connectivity is introduced to measure the network robustness with comparison to the popular average nodal degree metric.

## 1.2 Motivation

Despite the extensive prior research on network resiliency, the current approaches still have shortcomings due to the complex nature of SCA for survivable routing. Among different mechanisms for SCA, the weakness of the fixed-routing approach is the need for keeping up-to-date with the rapidly changing network state. Therefore, the performance of the fixed-routing based SCA technique is unacceptable in practice. On the other hand, the existing adaptive SCA algorithms that can provide optimal or near optimal routing solutions, require centralized and global routing information. This requirement has two problems:

- The communication networks must be equipped with a very complex network control layer which co-exists above the physical layer to continuously broadcast the link-state information toward every network node. This kind of centralized and global approach has fault-tolerance and scalability problems;
- The maintaining of such global routing information, either centralized at a command and control node or distributed among network nodes, may lead to an inaccurate routing information condition that can terribly degrade network performance when adaptive SCA algorithms determine a routing solution based on this inaccurate information [10-11].

In order to overcome the above disadvantages, a new approach for building a fully distributed SCA algorithm is needed. It must be efficient in terms of capacity usage and also be applicable to practical networks. Aiming at this target, we propose a novel Distributed Resilience Matrix (DRM) as a possible solution, which is inspired by the Spare Provision Matrix (SPM) structure [12]. In addition, there have been numerous publications on network routing with swarm intelligence since the first milestone paper, [13], in this area, but the ant-based routing algorithm e.g., in [14] for data communication networks can not be applied directly into the area of network resiliency. It is also notable that no results are available for applying the ant-based approach to solve the SCA problem. These observations motivate us to extend this promising heuristic mechanism on a new direction to solve the SCA problem and also to use the mobile agents to implement the updating of capacity usage information in the DRM structure.

Additionally, in our simulation studies of FoF-R routing algorithm with the DRM structure, the experimental results reveal that the average nodal degree metric, which we used to measure network connectivity, fails to adequately capture the network robustness characteristic in SCA. This finding motivates us to work on the network topology aspect to explore a more informative topological metric to measure network robustness. By introducing the algebraic connectivity metric, i.e., the second smallest eigenvalue of the Laplacian matrix of the network graph, our numerical results show that this metric is more accurate and informative than the average nodal degree metric.

### **1.3 Contributions**

This dissertation focuses on investigating the SCA problem for survivable routing. Our first contribution is a novel Distributed Resilience Matrix (DRM) framework, which is used to model the dependence between the working and spare capacities. This additional relationship information is critical to explore the sharing potential among protection paths in SCA problem.

The second contribution is the Friend-or-Foe Resilient ant-based routing algorithm. It is noted that the ant-based routing algorithms developed so far by other researchers in the communication area have concentrated on the coordination behaviour between ant agents. We postulate that network resiliency can benefit from augmenting ants' competitive behaviour with detestation behaviour between agents. This enrichment enables the ant agents to find disjoint routes from working paths, as alternative good quality paths to reroute the traffic upon failure and also to explore the capacity sharing potentials among their protection paths. Therefore, we have introduced both cooperation and detestation behaviors to the ants, adopting a "Friend or Foe" philosophy by using a capacity headroom-dependent attraction and repulsion function. The artificial ant is thus armed with ability for identifying who has already deposited the pheromones on the routes, whether it is a Friend or Foe. Then, an ant agent can make movements and deposit pheromones with different strategies. Through extensive simulations, we show that the FoF-R ant-based mobile agent approach can solve the SCA problem in survivable routing. It can reduce the connection setup delay for recovery and improve significantly the network performance in comparison with other algorithms.

Furthermore, in studies of network resiliency, it is important to be able to differentiate network topologies by means of a robust numerical measure that would indicate the levels of

immunity of these topologies to failures of their nodes and links. Ideally, such a measure should be sensitive to the existence of nodes or links which are more important than others, for example, if their failures cause the network's disintegration. Our third contribution is to introduce an algebraic connectivity metric to quantify the network robustness other than the average nodal degree metric, which is currently popular within the research community. The concept of algebraic connectivity is adopted from the spectral graph theory, and is the 2<sup>nd</sup> smallest eigenvalue of the Laplacian matrix of a given network. Extensive simulation studies confirm that this metric is more informative and accurate than the average nodal degree for characterizing the network topologies in studies of network resiliency.

### 1.4 Thesis Outline

The objective of this dissertation is to study the SCA problem for survivable telecommunication networks. It focuses on backbone networks in a single link or node failure scenario. The main task is to establish connections in the network such that there is an efficient SCA strategy to survive existing traffic flows upon any single failure in the network.

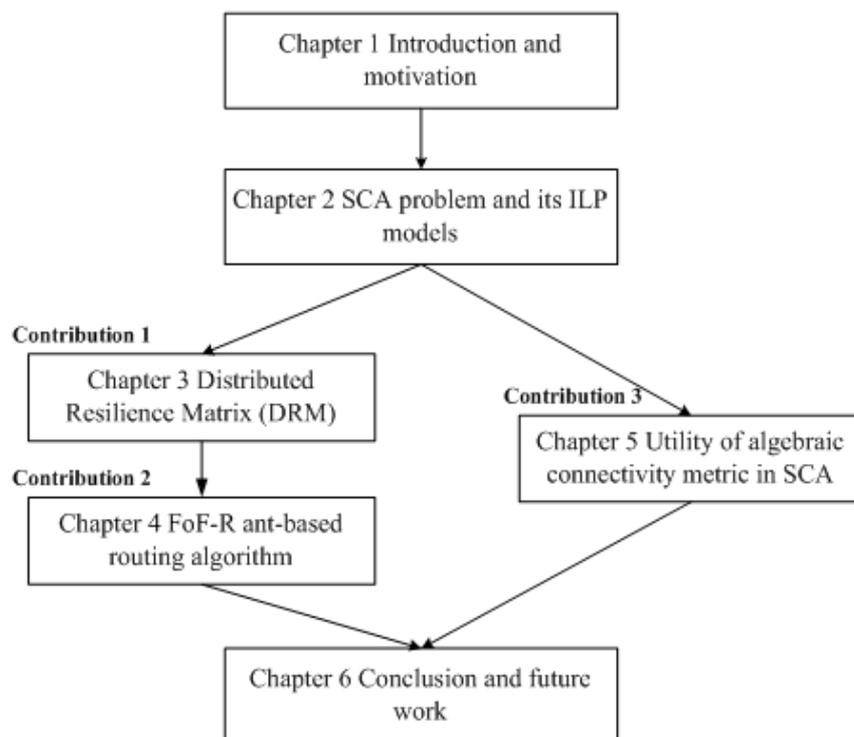


Figure 1.1 Thesis outline

This dissertation's structure is depicted in Figure 1.1. The remainder of this dissertation contains an introduction of relevant background knowledge, followed by benchmarking studies, and comprehensive analyses and discussion of three key advances in the area of network resiliency.

**Chapter 2** gives an introduction to network resiliency and reviews the current network resilience techniques and protection categories. The spare capacity allocation (SCA) problem is described and the related ILP models corresponding to the benchmarking SCA mechanisms such as Shared Backup Path Protection (SBPP) and p-cycles are described and discussed.

**Chapter 3** introduces background knowledge on capacity sharing. Then the existing Spare Provision Matrix (SPM) is described and the new matrix-based framework, named Distributed Resilience Matrix (DRM), is proposed to capture the dependence between the working and spare capacities in a distributed manner. This additional information is critical for exploring the capacity sharing potential among protection paths.

**Chapter 4** presents a novel ant-based routing algorithm, called the Friend-or-Foe Resilient (FoF-R) ant-based routing algorithm, to implement the distributed signaling for updating and exchanging the capacity usage information in DRMs framework. In addition, the FoF-R survivable routing algorithm can find the optimal protection cycle (i.e., two node-disjoint paths between a source-destination node pair) and explore the sharing ability among protection paths using a capacity headroom-dependent attraction and repulsion function. Simulation results based on the OMNeT++ tool show that the FoF-R scheme with the DRM structure is a promising approach to solve SCA problem for survivable routing and it gives a good tradeoff between solution optimality and computation speed.

**Chapter 5** develops deeper into network topology research related to network resiliency. It focuses on introducing a new topological metric i.e., algebraic connectivity to quantify the network robustness and also to find its correlation to the SCA. Instead of using the popular

average nodal degree metric to quantify the network robustness, we introduce a more informative metric i.e., algebraic connectivity, which is the second smallest eigenvalue of Laplacian matrix of a given network. We perform a thorough comparison of topologies to validate the new metric and extensive results presented reveal that the average nodal degree metric fails to capture the details of network robustness characteristic, whereas using algebraic connectivity metric, the network robustness properties can be better characterized.

**Chapter 6** summarizes the main contributions of this dissertation as well as explains the limitations of the research work presented. We also suggest some possible directions for future research such as extending the FoF-R routing algorithm's capability to handle arbitrary failure scenarios.

## **1.5 Published Papers**

William Liu, Harsha Sirisena and Krzysztof Pawlikowski, "FoF-R Ant: Ant-Based Survivable Routing Scheme for Shared Path Protection," in Proc. of Australian Telecommunication Networks and Applications Conference, 2008. ATNAC 2008.

William Liu, Harsha Sirisena and Krzysztof Pawlikowski, "FoF-R Ant-based Survivable Routing Using Distributed Resilience Matrix," in Proc. of 21st International Teletraffic Congress (ITC 21), Traffic and Performance Issues in Networks of the Future, September 15-17, 2009, Paris.

William Liu, Harsha Sirisena and Krzysztof Pawlikowski, "A Novel Resilience Matrix for Survivable Routing in a Distributed Control Architecture," in Proc. of 15th Asia-Pacific Conference on Communications (APCC2009), October 5-10, 2009, Shanghai.

William Liu, Harsha Sirisena and Krzysztof Pawlikowski, "Efficacy of Fiedler Value versus Nodal Degree in Spare Capacity Allocation," in Proc. of 15th Asia-Pacific Conference on Communications (APCC2009), October 5-10, 2009, Shanghai.

William Liu, Harsha Sirisena and Krzysztof Pawlikowski, “Weighted Algebraic Connectivity Metric for Non-Uniform Traffic in Reliable Network Design,” in Proc. of International Workshop on Reliable Networks Design and Modeling (RNDM2009), October 12-14, 2009, St. Petersburg.

William Liu, Harsha Sirisena and Krzysztof Pawlikowski, “Utility of Algebraic Connectivity Metric in Topology Design of Survivable Networks,” in Proc. of 7th International Workshop on the Design of Reliable Communication Networks (DRCN 2009), Washington DC, October 25-28, 2009.

William Liu, Harsha Sirisena and Krzysztof Pawlikowski, “A Novel Distributed Resilience Matrix for Arbitrary Failures in Spare Capacity Allocation,” in Proc. of 7th International Conference on Information, Communications and Signal Processing (ICICS 2009), December 7-10, 2009, Macau.

# Chapter 2

## Spare Capacity Allocation for Survivable Routing

In this chapter, we briefly discuss the fundamentals of network resiliency and then describe the spare capacity allocation (SCA) problem for survivable routing. Designing a low cost and survivable telecommunication network is an extremely complicated process and there is a growing consensus that mathematical programming belongs in the network designer's toolkit. The mathematical modeling coupled with optimization solvers has greatly reduced the burden of implementation of mathematical programming theory into the practice of network design. For benchmarking our proposed algorithm, we first present some Integer Linear Programming (ILP) models developed for solving the SCA problem.

The remainder of this chapter is organized as follows. First, we give some background knowledge on network resiliency and then the classification of protection schemes is introduced. The mathematical modeling for the SCA problem is presented and the ILP models for link protection, shared backup path protection (SBPP) and p-Cycles schemes are developed. Finally, critical information on the dependence between working and spare capacity in SCA for survivable routing is described.

### 2.1 Background on Network Resiliency

Network resiliency is the ability of the network to provide and maintain an acceptable level of service in the case of various failures of normal operation. It reflects the ability of a network to continue to function during and after failures. The following are key aspects of network resiliency that we need to consider in our studies.

### **2.1.1 Restoration and Protection**

Throughout the preceding discussions on the SCA problem, we have used the terms restoration and protection to refer to various network resilience mechanisms and actions interchangeably. However, in general, the term “protection” is used for schemes where the switching actions required after failure are pre-defined and spare capacity is often dedicated [15] to cover a specific set of failure scenarios. In a pure protection mechanism, even cross-connection is unnecessary when the signal is switched to the protection path; the protection paths are pre-configured into a pre-tested and ready-to-use state. However, the term protection is also used to refer to schemes such as Shared Backup Path Protection (SBPP) where the protection route is known ahead of time, but the capacity allocation and cross-connection remain to be accomplished in real time after failure has occurred.

Restoration, on the other hand, generally refers to the mechanisms where backup paths do not need to be pre-defined and a network-wide allocation of spare capacity is not dedicated to any specific failure but it is configured as needed to restore affected carrier signals after a failure has occurred [16]. In their purest form, restoration mechanisms determine backup paths, allocate spare capacity, and form the appropriate cross-connections all in real time as a response to a failure, either through a centralized mechanism or a distributed protocol. However, depending on the specific implementation, some restoration mechanisms can carry out pre-planning operations, and even some amount of pre-configuration is possible. Restoration by its very nature is adaptive to unexpected changes in the network state, and as such, will typically exhibit better availability than protection mechanisms.

### **2.1.2 Centralized vs. Distributed Protection**

According to the decision making and availability of information, we can divide protection schemes into two main categories: centralized and distributed. Centralized protection schemes use a centralized management system to perform the protection functions, such as failure detection, selection of backup route and redirection of flows to the established alternative path. A centralized scheme has the advantage of always getting all network information available, including topology and link capacity availability, even during failure, so it is capable of finding an optimal configuration such as minimizing the total amount of capacity allocated. However, a centralized approach requires the central database to be

continually updated, and there exists a single-point of failure of the central collection/management point. In addition, the recovery speed is relatively slow with the centralized scheme due to the communication delay between the centralized controller and other nodes, and the concentration of processing load on the centralized controller.

To alleviate the negative impact of the centralized control for protection, some proposals [17-18] consider distributed protection control. In a distributed protection scheme, each node in the network is capable of handling failures and making re-routing decisions. The distributed protection is more robust because it does not need a global view of the network state and thus is not vulnerable to a single point of failure, but it will not necessarily find an optimal configuration after a failure. Also, many network operators are wary of trusting their networks to such a distributed self-organizing process.

### 2.1.3 Link vs. Path Protection

We can further categorize protection mechanism by differentiating between localized and end-to-end path protection [19]. Localized protection is when backup paths are established between the end nodes of the failure link, whereas end-to-end schemes restore affected demands by constructing backup paths between their individual source and destination nodes. The backup paths should be completely disjoint from the pre-failure working paths. End-to-end protection schemes are typically able to make more efficient use of network resources, and so it tends to require less spare capacity than localized protection schemes.

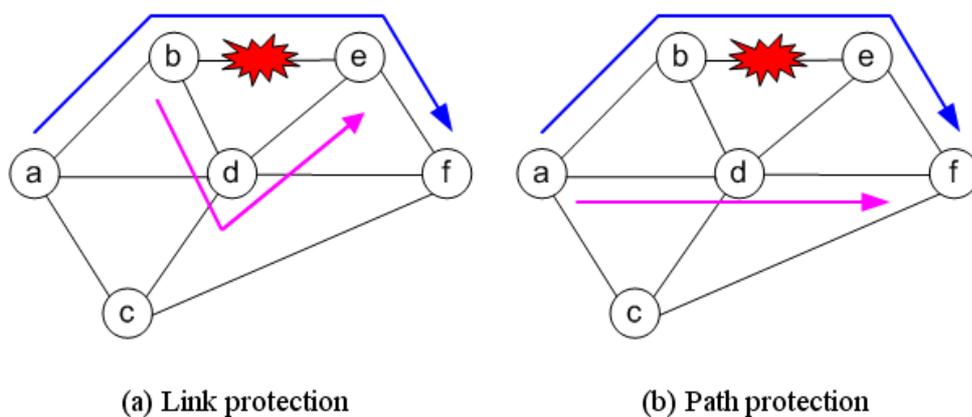


Figure 2.1 Link protection vs. Path protection

As shown in Figure 2.1, the most common form of localized protection is link protection, where a centralized or self-organizing re-routing mechanism deploys a

collectively coordinated set of backup paths between the end nodes of the failed link and effectively avoids the failure. Spare capacity is allocated at the time that the failure happens only, and will otherwise remain available for use as required to route protection paths for any failure scenario or carrying preemptible unprotected traffic. For working capacity on the failed link, one protection path must be established between the end nodes of the failed link, and so long as spare capacity is available to accommodate them. Link protection thus provides a logical detour comprised of a set of backup path segments around the break that disrupts working paths, without the consideration for the ultimate source-destination nodes of each working path being restored.

In path protection, the reaction to a failure takes place from an end-to-end viewpoint for each service path affected by the failure. Path protection schemes have several advantages. One is that they are more amenable to customer level control and visibility. Link protection is rather inherently a function of the transport network itself, whereas path level protection can be a function that is put in the users' control, or under the control of a service layer node such as a router. For example, in SBPP scheme, it is also possible for the users to know ahead of time, and even control, exactly where their service will be rerouted upon a failure. This is sometimes said to be important to customers. In addition, path protection spreads the overall rerouting problem more widely over the whole network and it is usually more capacity efficient than link protection, although the difference depend on network topology. In addition, path oriented protection schemes only require fault detection at the end nodes of the path, and this can be attractive for transparent or translucent optical networks where signal monitoring may not be available at intermediate nodes. On the other hand, path protection schemes are generally not as fast as their link oriented counterpart due the greater distances and numbers of nodes involved in recovery procedure signaling. Availability issues can also arise when working and protection paths are both long.

#### **2.1.4 Protection Techniques**

All types of network resilience techniques involve network spare resources allocation to the rerouting mechanisms. Based on the availability of network resources [20], protection mechanisms can be classified into the following four major types:

- 1+1 Protection: A dedicated backup path is predefined and the traffic is sent over both working and backup paths simultaneously. The receiver has some local algorithms to choose the best signal;
- 1:1 Protection: A dedicated backup path is predefined, the traffic is rerouted to the backup path only after the failure has occurred;
- 1: $N$  Protection: A dedicated backup path is predefined to protect a number of  $N$  working paths. The traffic will be switched to the backup path, if any of the working paths fail, but after that, the remaining  $N-1$  paths are unprotected;
- $M:N$  protection: The number of  $M$  dedicated backup paths are predefined to protect  $N \geq 1$  working paths, where  $M \in [1, N]$ .

The dedicated 1+1 protection is the fastest, as the traffic is being simultaneously transmitted over working and protection paths. However, compared to an unprotected system, it requires twice the amount of network resources, i.e., 100% redundancy. This technique has been widely used in Automatic Protection Switching (APC) of premium or high availability services. It should be noted that both 1:1 and 1:  $N$  protections are actually special cases of  $M: N$  protection technique. In  $M: N$  protection, the  $M$  backup paths are used to protect the  $N$  working paths. This provides a better utilization of resources than 1+1 protection, since backup paths are used by multiple working paths. Also, the idle backup resources can be used by low priority traffic to enhance further network resource utilization. However, this improved resource utilization in  $M: N$  is obtained at the cost of additional signaling and increased protection switching time, which will increase the overall recovery time of the network against faults.

### 2.1.5 Shared Risk Link Groups

Most of the previous protection mechanisms are designed for protection against failure of an individual or a diverse set of links, which may also have failure correlation to other components in a network. There exists a much broader concept known as Shared Risk Link Groups (SRLGs) for the design of survivable networks. An SRLG is a group of network links that share a common physical resource e.g., cable, conduit or node, whose failure will cause the failure of all links of the group. Thus, all links in the SRLGs have a shared risk of failure. SRLGs have been proposed as a fundamental concept for failure management in the Generalized Multi-Protocol Label Switching (GMPLS) control plane. In general, the information on SRLGs is obtained manually by the network operator with the knowledge of

the physical fiber plant of the network. This dissertation is limited to focus on single link or node protection and it does not include the consideration for the network protection with SRLGs. Extension of protection schemes involving SRLGs is discussed in [21], see also references there.

### 2.1.6 P-cycle

The concept of Preconfigured Protection Cycle (p-cycle) was introduced by Grover in [22-23]. P-cycles can be characterized as preconfigured protection cycles in a mesh network and it combines the speed of ring networks with the capacity efficiency of mesh networks [24-25]. P-cycles are ring-like pre-configured structures of spare capacity used to protect against failure of on-cycle links i.e., those links that are a part of the p-cycle, and straddling links i.e., those links whose end nodes are both on the p-cycle, but which are not actually a part of the p-cycle itself. Upon failure of a protected link, the p-cycle is “broken into” by the protection mechanism to re-route the traffic around the failure. The fundamental difference between p-cycles and rings, and the cause of p-cycles’ increased efficiency, is in the protection of straddling link failures. A unit-sized ring can only protect against the failure of a single wavelength on each link on the ring itself, but a unit-sized p-cycle can protect the failure of a wavelength on each on-cycle link as well as two wavelengths on each straddling link, for the same amount of spare capacity. The reason a p-cycle can protect two units of working capacity on each straddling link is because: if the straddling link fails, one unit can be restored in the clockwise direction around the cycle, and the other can be restored in the counter-clockwise direction.

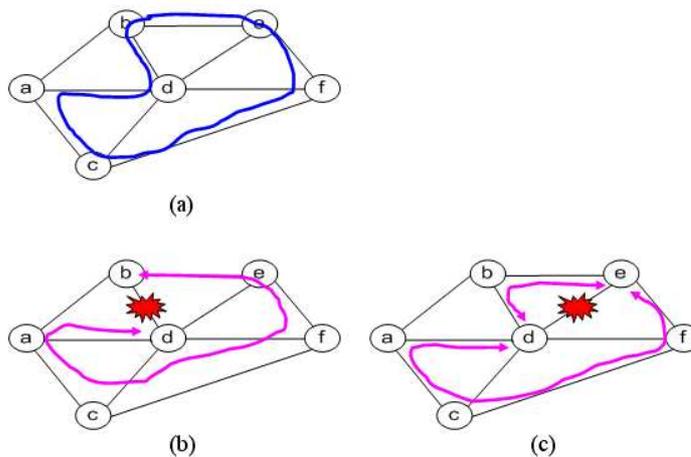


Figure 2.2 An example of p-cycle protection

An example of p-cycle protection in a network with 6 nodes is illustrated in Figure 2.2. The p-cycle is as shown in Figure 2.2(a). For failure of an on-cycle link in Figure 2.2(b), the working capacity on the failed link is re-routed around the p-cycle. For failure of a straddling link in Figure 2.2(c), the working capacity on the failed link can be re-routed in either direction around the p-cycle. If there are two units of working capacity on the failed straddling link, then one unit can be restored in each direction.

While it is not quite as capacity efficient as other mesh protection schemes e.g., link protection and SBPP, the p-cycles provides the same very rapid protection as rings because they are pre-connected prior to failure. P-cycles are significantly more efficient than rings because of their ability to protect straddling links. A simple comparison of a ring and a p-cycle will easily demonstrate how important the protection of straddling links is to p-cycle efficiency. P-cycles have another key benefit over rings, in that working paths can be shortest path routed through the network; they are not constrained by the p-cycle systems used to protect them. In a ring network, on the other hand, working path routing must follow the ring structures and inter-ring transitions. In fact, working paths on a straddling link do not even need to be routed within any p-cycle at all to be protected as long as each link crossed is a straddling link. Because of the advantages associated with p-cycle protection, we use it as one of benchmarking SCA mechanisms to our proposed algorithm. In addition, we use Shared Backup Path Protection (SBPP) as another benchmarking SCA mechanism, which is introduced in the following section.

### **2.1.7 Shared Backup Path Protection**

The shared backup path protection (SBPP) scheme is favored for optical networking and Multi-protocol Label Switching (MPLS) applications [26-27]. It is a path-oriented protection scheme with a particular combination of operational simplicity, speed, and efficiency that makes it of special interest for IP-centric optical network and MPLS layer contexts. The Open Shortest Path First - Traffic Engineering (OSPF-TE) and Constraint-based Label Distribution Protocol (CRLDP) type protocols enable service layer applications to set up their own SBPP path arrangements on demand. In addition, SBPP is a failure independent path oriented scheme where the protection route is identified in advance, but spare capacity has to be cross-connected on the backup route in real time. In other words, SBPP is an intermediate scheme between pure restoration and pure protection. It is perhaps best described as a preplanned restoration scheme. Any failure that affects the working path

causes a switchover to the predefined backup route and cross-connection to form a backup path. One can equivalently think of SBPP as a form of 1+1 APS dedicated path protection, while the capacity used to form the protection path is shared over failure of disjoint working paths.

Moreover, it might be difficult to know immediately where the failure has occurred. The failure condition may only be known at the end nodes of the paths affected. In this case, the SBPP scheme can decide in advance upon a set of end to end backup paths that are fully disjoint from each working path protected. The advantage is that it allows activation of the switchover by the end nodes without any other knowledge about the failure. For efficiency, the spare capacity is shared among backup paths that have disjoint working routes. For coordination of this capacity sharing, the global information on the relationship between the working and protection paths is required.

## **2.2 Spare Capacity Allocation Problem**

Compared to the restoration schemes which have no spare capacity pre-allocated before failure, in SBPP scheme, the pre-planning spare capacity mechanism not only guarantees service, but also minimizes the recovery time and range of the failure impact. This sort of service guarantees are especially important in packet-switching networks because backlog traffic accumulated during the failure recovery might introduce significant congestion [28-29] and it can be somehow mitigated by the pre-planning spare capacity mechanism. On the other hand, to reserve additional spare capacity will cause increased usage on network resources and degrade the network capacity for new traffic requirements under normal conditions. Thus, how to efficiently utilize spare capacity resources is becoming a more and more critical issue in network resilience design.

The SCA design can take the form of either a two -step or a single-step problem. In the two-step problem, working paths are routed first by some method and usually follow the shortest path routing, and then a spare capacity allocation (SCA) procedure optimally determines the restoration routing in the network. In the single step SCA design problem, working and protection paths routing, and subsequent working and spare capacity determination are performed jointly, so that the total network capacity cost is optimized. This joint optimization method is generally called Joint Capacity Allocation (JCA). This allows

working paths to be routed in other than a shortest path manner, so that in conjunction with the spare capacity needed for protection, the total capacity requirement is minimized.

There are various methods for implementing SCA and JCA design problems, and the exact model used will depend on the specific protection mechanism employed within the network. For instance, the SCA design problem for a link protection network is generally a form of non-simultaneous single-commodity capacity allocation problem [30] and early work on similar problems was to support time-varying network flow patterns. The main difference in applying that work to link protection is that we delete one link of the network for each of the non-simultaneous flow requirements, thereby simulating link failures. Other work that was done specifically for transport network restoration started with a proposed Linear Programming (LP) representation of the SCA based on a min-cut max-flow model [31]. Here, spare capacity is allocated so that for each possible link failure, the minimum spare capacity cutset on the surviving links is sufficient for full protection of the failed link's working capacity. In this context, they have defined a cutset as any set of links whose removal from the network would result in the end nodes of the failed link being in two disconnected components of the network. The minimum spare capacity cutset is the one whose links carry the minimum total spare capacity and it is shown that the maximum flow possible between any two nodes in a network is equivalent to the minimum spare capacity cutset. One technical challenge with this approach is that the number of possible cutsets in a network is  $O(2^L)$ , where  $L$  is the number of links, and so enumerating all cutsets becomes computationally infeasible. Thus, finding a suitably small subset of cutset that fully constrain the solution and also permit an optimal (or near-optimal) capacity design is difficult. Enhancements in [32] use an efficient algorithm to discover relevant new cutsets and a "path table" data structure to allow for fast restorability testing. For more information on the min-cut max-flow solution methods for the SCA problem, refer to [26, 31, 32].

In addition, a link-path LP model for the SCA problem in a link protection network was proposed in [33-34]. Here the network topology is first pre-processed to find all distinct logical routes that are feasible for use in the restoration of each failure scenario. Spare capacity values on each link are sized to support the largest assignment of simultaneous restoration flows over the feasible protection routes crossing each link in the network over all non-simultaneous failure scenarios such that the total spare capacity is minimized. The number of distinct routes possible is  $O(2^L)$ , but the complexity of the problem can be greatly reduced in practice by reducing the number of feasible routes provided to the problem,

through the use of route hop-limits to restrict route lengths with little or no loss of solution quality. This approach also gives a detailed explicit specification of restoration routes and flows, while the min-cut max-flow approach does not. Another practical advantage of the link-path method is that restoration route properties can be controlled to limit such properties as length, hop count, signal loss, etc., for each failure scenario, while the min-cut max-flow approach does not. The SCA design problem can also be expressed in the form of a set of transshipment or network flow LP problems [35]. In a network flow problem, supply nodes and demand nodes act as sources and destinations of a commodity (i.e., a path demand), and transshipment nodes simply act as intermediaries that pass along any of the commodity it receives to other nodes. Like the link-path model, the network flow model can explicitly specify the amount of flow over restoration routes, but it does not allow for easy control of restoration route properties.

The network SCA design methods used in this dissertation follow the link-path type of model. The exact structure of the models differs depending on the survivable routing mechanism we are using, but in general, all such models require an explicit enumeration of a set of feasible restoration routes and in the case of joint designs, working routes as well.

## **2.3 ILP models for SCA**

In this section, we shall discuss ILP models for protection schemes based on the abstract graph representation of the network. The degree of a node is the number of links incident to the node. A network is said to be 2-connected if there are at least two node-disjoint paths between any pair of nodes. The survivable network must be at least 2-connected. The problem of selecting a minimum cost set of links to ensure that a network is 2-connected is a fundamental problem in SCA design. We have modified the link-path models for link protection, SBPP and p-cycle SCA mechanisms based on some previous work [26, 36-38], which will be used to benchmark the SCA mechanism to our proposed heuristic solution and also to find the correlation between the topological metric and the total capacity allocated.

### **2.3.1 Link Protection**

The basic link-path model of the SCA problem in a link protection network uses the following notation:

**Sets:**

- $\mathcal{L}$ , the set of links in the network, and is typically indexed by  $i$  when referring to a failure link and  $j$  when referring to a corresponding protection link;
- $P_i$ , the set of all distinct feasible routes available to carry restoration flow for failure of link  $i$ , and is typically indexed by  $p$ .

**Input parameters,**

- $c_j$ , the cost of each unit of capacity (working or spare) on link  $j$ ;
- $w_i$ , the amount of working capacity to be protected on link  $i$ ;
- $\delta_{i,j}^p \in \{0,1\}$ , a parameter that encodes the protection routes. If  $\delta_{i,j}^p = 1$ , the route  $p$  is used for protection of link  $i$  and it traverses link  $j$ . If  $\delta_{i,j}^p = 0$ , the route  $p$  is used for protection of link  $i$  but it does not traverse link  $j$ .

**Decision variables:**

- $s_j \geq 0$ , the amount of spare capacity that is allocated on link  $j$ ;
- $f_i^p \geq 0$ , the amount of protection flow assigned to route  $p$  for failure of link  $i$ .

The ILP model for the link protection SCA is then expressed as follow,

$$\text{Minimize } \sum_{j \in \mathcal{L}} c_j \cdot s_j \quad (2.1)$$

$$\text{Subject to } \sum_{p \in P_i} f_i^p = w_i, \forall i \in \mathcal{L} \quad (2.2)$$

$$\sum_{p \in P_i} \delta_{i,j}^p \cdot f_i^p \leq s_j, \forall i, j \in \mathcal{L} | i \neq j \quad (2.3)$$

The objective function in equation (2.1) minimizes the total cost of spare capacity allocation in the network. We usually associate  $c_j$  with the length of the link i.e., the Euclidean distance between the end-nodes of the link, and in general, this cost can be regarded as representing the actual costs of fiber, cable etc. The constraints in equation (2.2) ensure that the total capacity assigned to all feasible protection routes for failure of link  $i$  is sufficient to fully protect whole working capacity on the failed link  $i$ . Equation (2.3) guarantees enough spare capacity on each protection link  $j$  to accommodate all restoration flows traversing it to protect failed link  $i$  for any  $i$ . In other words, each  $s_j$  quantity in equation (2.3) is determined by the largest sum of simultaneously imposed protection flows

on link  $j$ , over all non-simultaneous failure scenarios not involving link  $j$  itself as a failed component. Thus, the spare capacity assigned to each link  $j$  could arise from any of a number of different finite-flow sub-problems, there is such a sub-problem for each link failure scenario. Each individual failure scenario, taken in isolation, is similar to a two terminal minimum cost network flow problem, but the model above couples them all together under the objective of minimum spare capacity. It is for this reason that the constraints in equation (2.3) are not strict equalities; the spare capacity required on link  $j$  for one particular failure scenario may exceed that required for another failure scenario. The overall result of the model is a minimum sum of link-based maximum quantities of the protection flows assigned to each link.

In case of a WDM optical network where a unit of capacity represents an individual wavelength, the SCA problem could be solved as a pure ILP, with all  $s_j$  and  $f_i^p$  variables taking on strictly integer values only. The SCA problem can be solved as an LP for runtime considerations i.e., ILPs are much more difficult to solve than LPs, and rounding and variable adjustments were used to approximate the optimal integer solution. This can usually be done with only minor loss of optimality, as discussed in [39-40], as long as the capacity variables ( $s_j$ ) are integer, the integrality requirement on the underlying flow variables ( $f_i^p$ ) can be relaxed without affecting optimality or feasibility.

The SCA problem for a link protection network can easily be modified and formulated as the JCA problem in order to perform joint working and spare capacity optimization. To do so, the prior  $w_i$  input parameters become output variables, we need to modify the objective function, and add two new constraints to ensure the routing of working demands and adequate working capacity to support them. Let:

- $D$ , the set of working path demands, and is typically indexed by  $r$ .
- $Q^r$ , the set of all feasible routes available to carry working path for demand  $r$ , and is typically indexed by  $q$ .

In addition, we introduce the following new input parameters as:

- $d^r$ , the number of traffic demand units for demand  $r$ .

- $\xi_j^{r,q} \in \{0,1\}$ , a parameter that encodes working routes. If  $\xi_j^{r,q} = 1$ , the working route  $q$  is used for demand  $r$  that crosses link  $j$ . If  $\xi_j^{r,q} = 0$ , the working route  $q$  is used for demand  $r$  and it does not cross link  $j$ .

New decision variables are:

- $g^{r,q} \geq 0$ , the amount of working flow assigned to working route  $q$  used for demand  $r$ .
- $w_j \geq 0$ , the amount of working capacity that is allocated to link  $j$  (this was formerly an input parameter).

All previous notations used in equations (2.1) to (2.3) of the link protection network problem remain. In order to consider working capacities in addition to spare capacity, we make the following change to the objective function:

$$\text{Minimize } \sum_{\forall j \in \mathcal{L}} c_j \cdot (s_j + w_j) \quad (2.4)$$

Finally, we add two new constraint sets as follows:

$$\sum_{\forall q \in \mathcal{Q}^r} g^{r,q} = d^r \quad \forall r \in \mathbf{D} \quad (2.5)$$

$$\sum_{\forall r \in \mathbf{D}} \sum_{\forall q \in \mathcal{Q}^r} \xi_j^{r,q} \cdot g^{r,q} = w_j \quad \forall j \in \mathcal{L} \quad (2.6)$$

Equation (2.5) is the working path routing equivalent to the protection-related constraints in equation (2.2). It ensures that the total working flow assigned to all feasible working routes for demand  $r$  is sufficient to fully route it. Equation (2.6) sizes the working capacities on each link in much the same way that equation (2.3) does for spare capacity. The main structural difference between those two constraint sets is that in equation (2.6), the working flow for each demand is applied to each link at the same time, hence the double summation. When sizing spare capacity in equation (2.3), on the other hand, the protection flow is applied separately for each link failure scenario. Also, we use equality here, rather than an inequality as in equation (2.3). This is because the total working capacity is strictly equivalent to that required to carry all working path demands, which are all routed simultaneously. While spare capacity in equation (2.3) is sized to accommodate individual failure scenarios separately, some of which may require more or less spare capacity than others on any particular link. Besides the change to the objective function and the addition of the two new constraint sets, the constraints in equations (2.2) and (2.3) remain a part of the

JCA model. While the link protection SCA problem can also be solved using the JCA model, by simply providing only a single feasible working route e.g., the shortest for each demand, it is typically solved using the separate model provided earlier.

Many variations of these optimization problems are possible. The model with capacity only is one such version where the goal is to minimize the amount of working and/or spare capacity required, rather than the cost of such capacity. It can be modeled in two ways, either by simply setting all  $c_j$  capacity cost values to the same value ( $c_j=1, \forall j \in \mathcal{L}$  is typical), or removing the capacity cost parameter  $c_j$  from the objective function. In the latter case, equations (2.1) and (2.4) are replaced by equations (2.7) and (2.8), respectively:

$$\text{Minimize } \sum_{\forall j \in \mathcal{L}} s_j \quad (2.7)$$

$$\sum_{\forall j \in \mathcal{L}} w_j + s_j \quad (2.8)$$

Another common variation of the problem is one where the network does not necessarily have to provide full (i.e., 100%) protection for each failure scenario. By introducing a new input parameter,  $0 \leq \alpha_i \leq 1$ , which represents the proportion factor of working capacity on link  $i$  that must be protected in the event of failure for that link, we can replace equation (2.2) with equation (2.9) as below. The right hand side of this new equation now effectively requires sufficient protection flow over all protection routes to restore a specified proportion of the failed link's working capacity. If integrality is strictly asserted on the flow variable  $f_i^p$ , then it may also be necessary to change the equality in equation (2.9) to a weak inequality i.e., " $\leq$ ", since a fractional amount of required working capacity to be protected may result.

$$\sum_{\forall p \in P_i} f_i^p = w_i \cdot \alpha_i, \forall i \in \mathcal{L} \quad (2.9)$$

In equation (2.3), the spare capacity requirements are calculated as the summation of the products of protection flow and a  $\{0, 1\}$  binary parameter encoding whether a particular route is crossing a particular link, and similarly for equation (2.6) on calculating working capacity requirements. However, we can also encode whether a route crosses a link by declaring an additional set  $\mathcal{L}_q \subseteq \mathcal{L}$ , which is the set of links in route  $q$ , and it is just as easily generated as the  $\delta_{i,j}^p$  and  $\xi_j^{r,q}$ . Then,  $\sum_{\forall p \in P_j} \delta_{i,j}^p \cdot f_i^p$  is equivalent to  $\sum_{\forall p \in P_i | j \in \mathcal{L}_p} f_i^p$  .and

$\sum_{\forall r \in D} \sum_{\forall q \in Q^r} \xi_j^{r,q} \cdot g^{r,q}$  is equivalent to  $\sum_{\forall r \in D} \sum_{\forall q \in Q^r | j \in \mathcal{L}_q} g^{r,q}$ . We can then replace equations (2.3)

and (2.6) with equations (2.10) and (2.11), respectively as below,

$$\sum_{\forall p \in P_i | j \in \mathcal{L}_p} f_i^p \leq s_j \quad \forall i, j \in \mathcal{L} | i \neq j \quad (2.10)$$

$$\sum_{r \in D} \sum_{\forall q \in Q^r | j \in \mathcal{L}_q} g^{r,q} = w_j \quad \forall j \in \mathcal{L} \quad (2.11)$$

### 2.3.2 Shared Backup Path Protection

In the above models for link protection, only one set of feasible protection routes needs to be considered for any given failure scenario, since only a single link fails at any given time, and only one commodity requires rerouting. The model for SBPP, on the other hand, needs to provide for rerouting of multiple commodities simultaneously and this requires significant structural differences in the SBPP mathematical model compared to the link protection model. In addition to the notation from above, we add the following new sets:

- $Q^r$ , the set of all feasible routes that can be used either for working or protection path routing of demand  $r$ . It is typically indexed by  $p$  if we consider it for working path routing or  $b$  if we consider it for protection path routing.
- $Q_j^r \subseteq Q^r$ , the set of all feasible routes that can be used either for working or protection path routing of demand  $r$ , which traverses link  $j$ .
- $\mathcal{L}_q \subseteq \mathcal{L}$ , the set of links in route  $q \in Q^r$ , as described above.

In addition, we have the following decision variables:

- $x_b^r \in \{0,1\}$ , a variable that encodes the assignment of protection routes. If  $x_b^r = 1$ , then protection route  $b$  is used to protect demand  $r$ . If  $x_b^r = 0$ , then protection route  $b$  is not assigned to protect demand  $r$ .
- $y_p^r \in \{0,1\}$ , a variable that encodes the assignment of working routes. If  $y_p^r = 1$ , then working route  $p$  is assigned for routing of demand  $r$ . If  $y_p^r = 0$ , then working route  $p$  is not assigned for routing of demand  $r$ .

- $z_{p,b}^r \in \{0,1\}$ , a variable that jointly encodes the assignment of protection routes and working routes, and acts as a variable that encodes the product of  $x_b^r$  and  $y_p^r$ . If  $z_{p,b}^r = 1$ , then working route  $p$  and protection route  $b$  are both assigned for use by demand  $r$ . If  $z_{p,b}^r = 0$ , then at least one of working route  $p$  and protection route  $b$  is not assigned for use by demand  $r$ . If  $z_{p,b}^r$  did not exist as a separate variable, then equation (2.19) below would need to contain a product of two variables, which would make this model non-linear.

Therefore, the SBPP JCA model can be expressed as follows:

$$\text{Minimize } \sum_{\forall j \in \mathcal{L}} c_j \cdot (s_j + w_j) \quad (2.12)$$

$$\text{Subject to: } \sum_{\forall p \in \mathcal{Q}^r} y_p^r = 1 \quad \forall r \in \mathbf{D} \quad (2.13)$$

$$\sum_{\forall r \in \mathbf{D}} \sum_{\forall p \in \mathcal{Q}_j^r} y_p^r \cdot d^r = w_j \quad \forall j \in \mathcal{L} \quad (2.14)$$

$$\sum_{\forall b \in \mathcal{Q}^r} x_b^r = 1 \quad \forall r \in \mathbf{D} \quad (2.15)$$

$$x_q^r + y_q^r \leq 1 \quad \forall r \in \mathbf{D}, \forall q \in \mathcal{Q}^r \quad (2.16)$$

$$x_b^r + y_p^r \leq z_{p,b}^r + 1 \quad \forall r \in \mathbf{D}, \forall p, b \in \mathcal{Q}^r \quad (2.17)$$

$$\sum_{\forall b \in \mathcal{Q}^r | i \in \mathcal{L}_b} x_b^r \geq y_p^r \quad \forall i \in \mathcal{L}, \forall r \in \mathbf{D}, \forall p \in \mathcal{Q}_i^r \quad (2.18)$$

$$\sum_{\forall r \in \mathbf{D}} \sum_{\forall b \in \mathcal{Q}_j^r} \sum_{\forall p \in \mathcal{Q}_i^r | \mathcal{L}_p \cap \mathcal{L}_b = \emptyset} z_{p,b}^r \cdot d^r \leq s_j \quad \forall i, j \in \mathcal{L} | i \neq j \quad (2.19)$$

The objective function (2.12) minimizes the total cost of working and spare capacity needed to provide working and protection paths routing of all demands, which is the same objective function as that for the link protection JCA model in equation (2.4). The constraints set in equation (2.13) force that there is exactly one working route for each demand  $r$ . Equation (2.14) determines the amount of working capacity required on each link to accommodate the working path routing; the working capacity on a link is equivalent to the sum of the number of working demand units of each demand  $r$ , whose working route  $p$  crosses the link. Equation (2.15) ensures that there is exactly one protection route assigned

for each demand  $r$ . In equation (2.16), any given route  $q$  can only be assigned as the working route or the protection route for a demand  $r$  (or neither), but not both. While this constraint is not strictly required because later constraints will ensure link disjointedness between any working and protection routes, this added-value constraint helps to more directly confine the feasible solution space of the ILP. Equation (2.17) assigns values to the  $z_{p,b}^r$  variables, which are used in the constraints set that follows. If  $x_b^r=0$  and  $y_p^r = 0$ , i.e., neither routes  $p$  nor  $b$  are assigned as the working and protection route, respectively, for demand  $r$ , then  $z_{p,b}^r$  will be allowed to equal 0, and since  $z_{p,b}^r = 0$  can only ever decrease spare capacity costs via the combination of other constraints in the model, then that is the value it will take. On the other hand, if  $x_b^r=1$  and/or  $y_p^r = 1$ , i.e., at least one of routes  $p$  or  $b$  are assigned as the working and protection route, respectively, for demand  $r$ , then  $z_{p,b}^r$  will have to equal 1 for the constraint to be satisfied. Effectively, this is equivalent to  $z_{p,b}^r = x_b^r \cdot y_p^r$ , but since that is a non-linear equation, then we must express it as we have; its purpose will become apparent in the discussion of equation (2.19), below. The constraints in equation (2.18) ensure that for each route  $p$  for demand  $r$  that crosses failed link  $i$ , if that route is assigned as the working route (i.e.,  $y_p^r = 1$ ), then that demand must be assigned a protection route  $b$  that does not cross the failed link (i.e.,  $i \notin \mathcal{L}_b$ ). Finally, in equation (2.19), sufficient spare capacity is assigned to each surviving link  $j$  to accommodate all protection route assignments that simultaneously cross that link for each failure scenario. More specifically, the spare capacity on protection link  $j$  must equal or exceed the sum of the number of path units of all demand  $r$  that are assigned protection route  $b$  which crosses link  $j$ , and working route  $p$  which crosses failed link  $i$ , thus routes  $p$  and  $b$  are disjoint, i.e.,  $\mathcal{L}_p \cap \mathcal{L}_b = \emptyset$ . If route  $p$  is not assigned as the working route for demand  $r$  and/or route  $b$  is not assigned as the protection route, then from equation (2.17),  $z_{p,b}^r = 0$ , and so the only combination of working and protection routes that will contribute to the spare capacity of link  $j$  is the pair that is actually assigned.

As in the link protection problem, we can also enforce integrality on the  $w_j$  working capacity and  $s_j$  spare capacity variables to correspond to WDM optical networking where a unit of capacity represents an individual wavelength. We must also enforce integrality on

the  $x_b^r$ ,  $y_p^r$  and  $z_{p,b}^r$  variables since they must strictly be either equivalent to 0 or 1 in order to have any meaning. Since there are no other output variables in this model, we must therefore solve the SBPP JCA design problem as a pure ILP. In much the same way that the link-protection SCA design problem could be solved by using the link-protection JCA model, we can solve the SBPP SCA design problem with only a simple modification to the SBPP JCA model. To do so, values for the  $y_p^r$  variables encoding whether an route  $p$  is a working route for demand  $r$  or not are assigned as inputs so that  $y_p^r=1$ , for example, the shortest route  $p$  assigned to be the working route for demand  $r$ , and  $y_p^r = 0$  for all other routes for demand  $r$ . Equation (2.13) becomes redundant but still holds, and all other constraint sets remain unaffected.

In these circumstances, it would also be wise to ensure that route sets provided for each demand contain only those routes that are link disjoint from the working route, so as not to unnecessarily add to the complexity of the problem by including routes and all the accompanying parameters, variables, and constraints that could not possibly be chosen as protection routes. Under the same objective of minimizing working and spare capacity costs, the model determines which pair of routes each demand is assigned for working and protection paths routing, subject to the same sets of constraints above. In other words, there is only a single working and protection route per demand, and spare capacity is sufficient to accommodate all simultaneously required protection routes for each link failure scenario. More details on this design model can be found in [26].

### 2.3.3 P-Cycles

Another ILP model for SCA for survivable routing that has been developed here is p-cycle protection. The p-cycle design problem differs structurally from the link protection and SBPP schemes in one key fashion: the optimization is in the form of a choice between feasible cycles rather than a choice between feasible routes. As such, we need to introduce new notation to properly encode cycles and to identify straddling links and on-cycle links. In addition to the notation used in previous models, we add the following new sets:

- $C$ , the set of feasible cycles that can be used to provide p-cycle protection for working links in the network and is indexed by  $p$ .

We also have the following parameters:

- $x_{p,j} \in \{0,1,2\}$ , a parameter equivalent to the number of protection relationships provided by a unit-sized copy of p-cycle  $p$  for working links on link  $j$ . Recalling that a p-cycle provides protection for one working link on an on-cycle link and two working links on a straddling link. Thus,  $x_{p,j}=1$  if link  $j$  is an on-cycle link for cycle  $p$ ,  $x_{p,j}=2$  if link  $j$  is a straddling link for cycle  $p$ ,  $x_{p,j}=0$ , if link  $j$  is neither an on-cycle or straddling link for cycle  $p$ .

Also, a new decision variable is added:

- $n_p \geq 0$ , the number of unit-sized copies of p-cycle  $p$  placed in the network.

Therefore, the full ILP model of the joint capacity allocation of p-cycle protection design is as follows:

$$\text{Minimize } \sum_{\forall j \in \mathcal{L}} c_j \cdot (s_j + w_j) \quad (2.20)$$

$$\text{Subject to } \sum_{\forall q \in Q^r} g^{r,q} = d^r \quad \forall r \in \mathbf{D} \quad (2.21)$$

$$\sum_{\forall r \in \mathbf{D}} \sum_{\forall q \in Q^r | j \in \mathcal{L}_q} g^{r,q} = w_j \quad \forall j \in \mathcal{L} \quad (2.22)$$

$$\sum_{\forall p \in \mathbf{C}} x_{p,j} \cdot n_p \geq w_i \quad \forall j \in \mathcal{L} \quad (2.23)$$

$$\sum_{\forall p \in \mathbf{C} | x_{p,j}=1} n_p = s_j \quad \forall j \in \mathcal{L} \quad (2.24)$$

Equations (2.20), (2.21), and (2.22) are identical to prior equations seen in link-protection JCA design and SBPP JCA design. Equation (2.20) minimizes the cost of working and spare capacity required in the network, while equations (2.21) and (2.22) ensure sufficient working flows to carry all demands, and calculate the working capacity required to accommodate those working flows, respectively. The constraint set in equation (2.23) ensures that for the failure of any link  $i$ , the p-cycles placed in the network provide enough protection relationships to fully reroute all of the working capacity on the failed link. Note that if  $x_{p,j} = 1$ , then p-cycle  $p$  will provide protection for one of the working links on link  $i$

per copy of the p-cycle, if  $x_{p,j}=2$ , then p-cycle  $p$  will provide protection for two of the working links on link  $i$  per copy of the p-cycle, and if  $x_{p,j} = 0$ , then p-cycle  $p$  will not provide any protection for working links on link  $i$ . Equation (2.24) determines the amount of spare capacity on each link  $j$  as a result of the p-cycles placed in equation (2.23) that cross that link. The only spare capacity optimization version of the above problem can be easily modeled by simply removing the constraint sets in equation (2.21) and (2.22), turning the  $w_j$  decision variables into input parameters with values e.g., the shortest path routing of demands, and also removing  $w_j$  from the objective function.

## 2.4 Feasible Routes Enumeration in ILP models

All of the ILP models developed in Section 2.3 are not limited to any particular network. They are applicable to any network by using appropriate pre-processing methods to fully express the design problems with the proper sets of feasible routes and/or cycles required. In order to obtain the strictly optimal solution to one of the design problems, the feasible route and/or cycle sets need to contain all distinct routes and/or cycles in network. However, for even moderately sized networks, such route sets and cycle sets are very large. A common and practical approach to deal with this issue is to set a hop limit, denoted as  $H$ , on the feasible routes, such that only those routes have  $H$  or fewer links are considered as feasible routes. The thinking is that since using shorter routes will require less capacity than using longer routes, then providing the problem with only short routes should allow a relatively efficient design. However, it is clear that as  $H$  is increased, more sharing-efficient patterns of re-routing are permitted, which is demonstrated in [34]. This allows for a fairly good trade-off between problem complexity and solution quality. A large  $H$  provides a greater number of routing options for the problem to choose from thereby reducing overall design costs but increasing the complexity of the problem i.e., the greater the number of feasible working and/or protection routes, the greater the number of constraints and variables in a design problem. A smaller  $H$ , on the other hand, restricts the number of routing options available. Although the complexity of the problem may be greatly reduced, but it is forced to select slightly less efficient and possibly more capacity will be allocated. As also shown in [34], there is some threshold hop limit,  $H^*$ , at which the theoretical minimum of capacity requirements is reached. In other words, the solution obtained when using no hop limit at all

is not better than that obtained when using a hop limit of  $H^*$ . We note that the exact value of  $H^*$  is dependent on a variety of factors, including the network topology, demands, and link costs, and cannot be pre-determined analytically. Other options for restricting the number of feasible routes is to use distance limits, cost limits, or even optical path loss limits, all of which will provide trade-offs similar to that provided by a strict hop limit. One practical problem comes when a network contains sparsely connected regions with long chains and/or low nodal degree as well as other more richly connected regions with higher nodal degree. In this case, a hop limit of  $H = 10$ , for example, may be necessary to provide even a single feasible protection route for some links in a sparsely connected region, while using that same hop limit might produce many thousands of feasible protection routes in a richly connected region of the same network.

In this dissertation, we use a related strategy that is both effective and practical at representing and solving the design models, and also greatly improves the scalability of the problem to permit solution of quite large network design problems with a variety of richly and sparsely connected regions. The idea is not to presume a specific network-wide hop or length limit and attempt to generate all distinct routes up to that limit, but rather to use a procedure that results in a specified number of the shortest distinct feasible routes at whatever hop or length limit is required independently of one another. So for instance, if we specify that at least 10 feasible protection routes for each demand, then the procedure is to enumerate the 10 shortest distinct protection routes for it separately.

## **2.5 Capacity Sharing Information in SCA**

All of the above models are assumed to applying centralized preplanning routing. In addition, routing method that picks the shortest backup path can also provide an approximate solution. Such methods have been extensively used for network design problems including SCA. The advantage for this algorithm is the fast solution speed but optimality is sacrificed and further adjustment are required for solution improvement as shown in [41]. The major bottleneck for routing based SCA algorithm is whether the spare capacity is shared among different failures, since such sharing can significantly reduce network redundancy, which has been extensively studied recently.

The Sharing with Partial routing Information (SPI) and Sharing with Complete routing Information (SCI) were introduced by [42]. In SPI, the backup path routing is based on the shortest path algorithm while the resource minimization is approximated by using modified link costs in routing. Although SPI is simple and fast, as shown in their numerical results, the redundancy obtained by SPI is not very close to the optimal results. While for a centralized SCI scheme, per-flow based information is necessary to find optimal results. Some other research efforts on this sort of routing-based algorithm for allocating spare capacity and backup paths can be found in [43-45]. The online algorithm introduced in [43] uses the shortest path algorithm to route a backup path for each flow sequentially. One of its differences from other routing based algorithms is that the routing link metrics are based on so called buckets. Each bucket of a link maintains the maximum spare capacity required on this link when one of the other links fails. The link metric is calculated based on a non-linear function of these buckets. Moreover, the Distributed Partial Information Management (DPIM) scheme [46-47] was proposed, in order to reduce the state information complexity from  $O(L^2)$  of the spare provision matrix down to  $O(L)$  with a trade-off in solution optimality. The proposed information collection process only needs to collect partial information for the spare capacity sharing in order to have less complexity.

The centralized SCA algorithms will result in slow restoration response times due to the network state information collecting and dispatching process. The distributed algorithms introduced so far, encounter scalability problems for flow numbers and network sizes and therefore still cannot achieve high resource efficiency. An efficient and fast spare capacity allocation (SCA) algorithm which can be implemented in a distributed fashion is needed. This strongly motivates us to seek a new solution, to be referred as the Distributed Resilience Matrix (DRM) framework to tackle the above difficulties.

## 2.6 Summary

In this chapter, we gave an overview of network resilience techniques and then described the spare capacity allocation (SCA) problem for survivable routing. An in-depth description of the spare capacity allocation (SCA) problem was provided and the ILP models for link protection, p-cycles and SBPP mechanisms were developed and discussed. Essential information on the dependence between working and spare capacity in SCA, which is

essential for exploring the sharing potentials among protection paths for survivable routing, was also introduced.



# Chapter 3

## Distributed Resilience Matrix

Fast recovery from failures, including efficient allocation of capacity in a network for guaranteeing seamless communication services, is the primary goal of spare capacity allocation (SCA) design in survivable routing. The level of capacity sharing that can be achieved for a given network depends on how detailed is the capacity usage information available at the source nodes, which is used to identify the dependencies between the working and protection capacities associated with each pair of links. In this chapter, we investigate such dependencies based on a new SCA structure, called the Distributed Resilience Matrix (DRM) under a distributed control environment.

The remainder of this chapter is organized as follows. First, we introduce some background knowledge on capacity sharing. The Spare Provision Matrix (SPM), which gracefully models the dependencies between working and protection capacity is introduced and then the novel Distributed Resilience Matrix (DRM) is presented. We also compare the DRM with the SPM to show its advantages for capturing network capacity usage information in a distributed manner. Finally, the essential signaling functions to update and exchange capacity usage information in DRM are described.

### 3.1 Information on Capacity Sharing

In the SBPP mechanism, the amount of capacity sharing that can be achieved among the protection paths depends on the capacity usage information available at the source nodes. There are three scenarios of available information to be considered. They can be categorized as the cases of minimal, partial and complete information in [48]. In the Sharing with

Minimal Information (SMI), the only information that the source node has is the residual capacity on each link. Therefore, it is not possible to do any sharing among the protection paths since the information for setting up shared paths is not available. In the Sharing with Partial Information (SPI), the source node has access to more information. Namely, it knows the total amount of capacity used by the working and protection paths, respectively. The SPI is fairly modest in terms of the amount of information to be maintained, and because only aggregate information is needed, it is easy to maintain it in a distributed fashion. In the last and ideal scenario, i.e., Sharing with Complete Information (SCI), the source node has complete information and it knows the route details for all the current traffic flows. While this sort of routing needs to be implemented in a centralized manner and it would be very difficult to disseminate complete routing information to all the nodes as the network size grows significantly. Among these three scenarios, SCI promises the best sharing of resources but implies a large overhead and it features poor scalability for large networks. Therefore, we propose the Distributed Resilience Matrix (DRM) to implement the SCI under a distributed control environment.

### 3.1.1 Background on Capacity Sharing

Given a network topology and traffic flows, the objective of the SCA mechanism is to optimize the total cost of network capacity allocated. The decision variables include where to route working and protection paths, and how much spare capacity should be reserved. The protection paths can share the capacity among common links if their corresponding working paths are not subject to the same failure. This additional information on the relationship between working and protection paths is critical for exploring the capacity sharing potentials in the network.

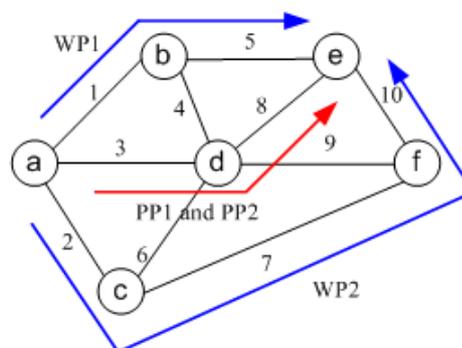


Figure 3.1 An example of capacity sharing in a simple network with 6 nodes

An example of capacity sharing in a simple network with 6 nodes is shown in Figure 3.1 above. We assume that there are two traffic flows between nodes “a” and “e” with capacity requirements  $b_1$  and  $b_2$  by using two disjoint working paths (WPs) i.e., WP1= (1, 5) and WP2= (2, 7, 10), respectively. Under the single failure scenario, the two WPs cannot fail at the same time. Accordingly, their corresponding protection paths will not to be used at the same time. Therefore, their two corresponding protection paths (PPs), i.e., PP1 and PP2 can all use path (3, 8) and they can share the protection capacity on link 3 and link 8 without affecting the survivability of either connection. The total reserved protection capacity that needs be allocated on links 3 and 8 for the two WPs is  $\max \{b_1, b_2\}$  rather than  $b_1 + b_2$ .

In the case of a new incoming flow  $r$  uses link  $i$  in its protection path, the capacity along link  $i$  can be categorized into three types as shown in Figure 3.2 below.

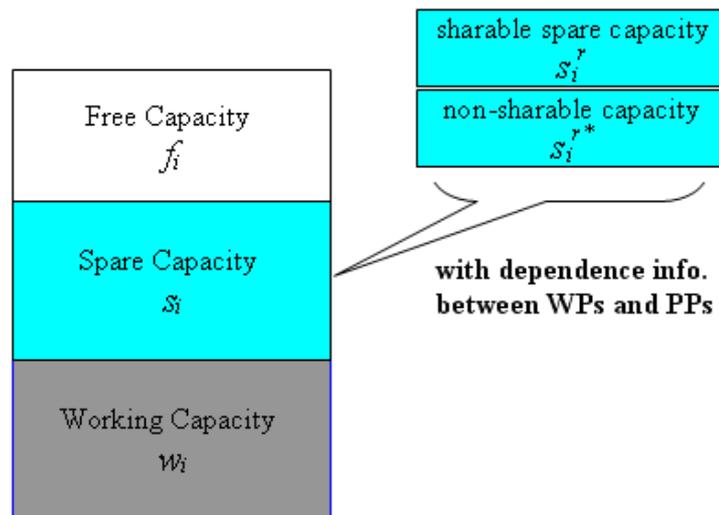


Figure 3.2 An illustration of capacity categories in link  $i$

- **Free capacity**, denoted as  $f_i$ , which is the link capacity that can be allocated as either working or spare capacity;
- **Working capacity**, denoted as  $w_i$ , which is the capacity occupied by the existent working paths and cannot be taken for any other use until the corresponding working paths are torn down.

- **Spare capacity**, denoted as  $s_i$ , which is the link capacity reserved by the existing protection paths. With knowledge of the relationship between the working and protection paths, the spare capacity in link  $i$  can be further categorized into the following two types according to the new coming flow  $r$ .
  - ❖ **Sharable spare capacity**, denoted as  $s_i^r$ , which is the link capacity that has been occupied by some other protection paths, but it is still sharable to traffic flow  $r$ , because their working paths are disjoint.
  - ❖ **Non-sharable spare capacity**, denoted as  $s_i^{r*}$ , which is the link capacity that has been occupied by some protection paths and it is not sharable to traffic flow  $r$  because their working paths overlap somewhere.

The information about the relationship between the working and protection paths in the SCA structure has been investigated by a number of researchers. The fault management table (FMT) method is the foundation for the Resource Aggregation Fault Tolerant (RAFT) scheme [49]. It provides a local data structure to store the spare capacity sharing information among different flows. It is very difficult to use the FMT to share this information globally since such information is per-flow based and hence it is not scalable with network size and number of flows. In addition, a two-dimensional array between failed link and link with spare capacity is proposed in [50] and the spare capacity can be also found using this array. Then the routing algorithm can use part of the information to build routing metrics to route protection paths. The “channel dependency graph” was introduced in [51] to analyze network fault tolerance when failure protection routes exist in a parallel computing system. Though it concentrates on the question of how many faults that a fault tolerant routing function can deal with, the dependency relation between links on working and protection paths is shown through a dual graph.

In addition, the backup load distribution matrix (BLDM) was introduced in [52], which can capture the partial network state and greatly reduces the amount of routing information maintained and exchanged. The spare provision matrix (SPM) has been proposed and further developed in [53-56], which is a milestone structure that allows concise modeling of dependencies between working and protection capacities based on a matrix form. In addition, an improvement to the link-state information dissemination process, called Sharing with

Reduced Information (SRI) [57-58] was introduced, it takes the advantage of the Singular Value Decomposition (SVD) technique [59] performed at each node for increasing the precision of SPM reconstruction. Their study has shown much improvement in the precision of SPM reconstruction.

From the above discussion, we can see that the SCA is a challenging combinatorial optimization problem and its structure is still under study [60]. Therefore, instead of using the estimation method or the complex SVD transformation to reconstruct the SPM or its similar structure BLDM in a distributed way, we propose a new Distributed Resilience Matrix (DRM) structure to decompose the SPM into the local dependency table for each link radiating from individual nodes. It significantly mitigates problems related to the difficulty of precise estimation of sharable spare capacity.

### 3.1.2 The SPM structure

The previous studies on dynamic survivable routing considered that, under the sharing with the complete information scenario (SCI), the knowledge of all the existing working and protection paths in the network can be concisely modeled using the spare provision matrix (SPM) introduced in [53]. In the SPM structure, a network is represented by an undirected graph with  $N$  nodes,  $L$  links and  $R$  flows. A traffic flow  $r$ ,  $1 \leq r \leq R$ , is specified by its source and destination node pair  $(o(r), d(r))$  and traffic demand  $d_r$ , here we assume only 1 unit of traffic demand for each flow, so  $d_r = 1$  for all  $r$ .

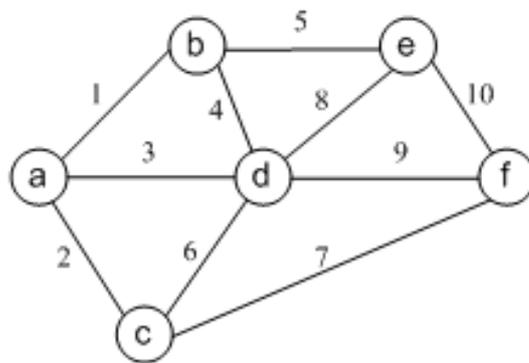


Figure 3.3 An example network of 6 nodes and 10 links.

Flow $r$	$o$	$d$	Working Path Matrix ( $P$ )										Protection Path Matrix ( $Q$ )									
			1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
1	$a$	$b$	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
2	$a$	$c$	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
3	$a$	$d$	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
4	$a$	$e$	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0		
5	$a$	$f$	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0		
6	$b$	$c$	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
7	$b$	$d$	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
8	$b$	$e$	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0		
9	$b$	$f$	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0		
10	$c$	$d$	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0		
11	$c$	$e$	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	1	0		
12	$c$	$f$	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0		
13	$d$	$e$	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0		
14	$d$	$f$	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0		
15	$e$	$f$	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0		

Figure 3.4 An example of SPM structure: matrices  $P$  and  $Q$ 

As shown in Figures 3.3 and 3.4, in an SPM structure, the working and protection paths of flow  $r$  are represented by two  $1 \times L$  binary row vectors  $p_r = \{p_{rl}\}$  and  $q_r = \{q_{rl}\}$ , respectively. The  $l$ -th element in one of the vectors equals “1” if and only if (iff) the corresponding path uses link  $l$ . The path-link incidence matrices for working and protection paths are the collections of these vectors, forming two  $R \times L$  matrices  $P = \{p_{rl}\}$  and  $Q = \{q_{rl}\}$  respectively. Let  $D = \text{Diag}(\{d_r\}_{R \times 1})$  denote the diagonal matrix representing the demands of flows. If the protection level of flows is under 100%, the elements in  $D$  can be adjusted to reserve partial spare capacities on protection paths.

$$G = Q^T D P \quad (3.1)$$

$$s = \max_{row} G \quad (3.2)$$

Fail link  $j$

→

	1	2	3	4	5	6	7	8	9	10	$s$
1	0	0	0	1	0	0	0	0	0	0	1
2	0	0	2	0	0	0	0	0	1	0	2
3	2	1	0	1	1	0	0	0	0	0	2
4	2	1	0	0	2	0	0	0	0	1	2
5	0	0	0	0	0	0	0	0	0	0	0
6	1	2	1	0	0	0	1	0	1	0	2
7	0	0	1	0	0	2	0	1	2	0	2
8	1	0	0	0	2	0	0	0	0	1	2
9	0	0	0	0	1	1	1	1	0	2	2
10	0	0	0	0	0	1	0	2	0	0	2

↓

Link  $i$

Figure 3.5 An example of Spare Provision Matrix (SPM)  $G$  and column vector  $s$

As shown in Figure 3.5, given the protection paths matrix  $Q$ , demand matrix  $D$ , and working paths matrix  $P$ , the spare provision matrix  $G$  can be determined by equation (3.1). Here it uses  $G = \{g_{lk}\}_{L \times K}$  denote the SPM whose elements  $g_{lk}$  are the minimum spare capacity required on link  $l$  when single link  $k$  failed. The minimum spare capacity required on each link is denoted by the column vector  $s = \{s_l\}_{L \times 1}$  which is calculated in equation (3.2). The function  $max$  in equation (3.2) asserts that any element in the spare capacity vector  $s$  is equal to the maximum number in the corresponding row of  $G$ . The row max operation guarantees that spare capacity in  $s$  is large enough to cover the capacity required by different failures. In this way, the minimum spare capacity on a link is always equal to the maximum spare capacity required by any single link failure.

While another way to calculate  $G$  instead of the SPM structure is to aggregate per-flow based information of working and protection paths. The contribution of a traffic flow  $r$  to  $G$  is given by  $G^r = \{g_{lk}^r\}_{L \times L}$ , where  $p_r$  and  $q_r$  are the row vectors. Thus, the spare provision matrix  $G$  can be also calculated in an aggregated version from:

$$G^r = d_r (q_r^T p_r), \quad \forall r, 1 \leq r \leq R \quad (3.3)$$

$$G = \sum_{r=1}^R G^r \quad (3.4)$$

### 3.2 Proposed DRM structure

In the SPM structure introduced above, the minimum spare capacity reserved on a link is always equal to the maximum spare capacity required by any single link failure and the maximum extent of spare resource sharing is achieved by assuming that each node must have per-flow information to acquire the matrix  $P$  and  $Q$ . In this SCI scheme, each node needs to broadcast information about its engagement in each flow. This yields a dissemination complexity  $O(RL)$ , making this scheme impractical in a distributed control environment. To tackle this problem, we propose a new structure named Distributed Resilience Matrix (DRM), which bears the same complete dependency information to obtain the vector  $s$ . In addition, it is done only by using simplified additive operations, instead of the multiplicative operations needed for SPM, and the complexity of the information exchange decreases to  $O(L)$ .

In DRM structure, we let  $T = D \cdot (P - Q)$  denote the complete DRM. Its negative entries represent protection capacity, while positive entries mean working capacity. In each row, we can identify the capacity used and the links traversed by the WP (if it is positive) and PP (if it is negative) for each row associated with its corresponding flow. In each column, the sum of positive entries represents the total amount of working capacity used by traffic flows in that link and the sum of negative entries represents the capacity reserved as protection capacity in each link by other WPs. An example of the complete  $T$  is illustrated in Figure 3.6 below.

	1	2	3	4	5	6	7	8	9	10
<b>Distributed Resilience Matrix</b> $T = D(P - Q)$	1	0	-1	-1	0	0	0	0	0	0
	0	1	-1	0	0	-1	0	0	0	0
	0	-1	1	0	0	-1	0	0	0	0
	1	0	-1	0	1	0	0	-1	0	0
	0	-1	1	0	0	0	-1	0	1	0
	1	1	0	-1	0	-1	0	0	0	0
	-1	0	-1	1	0	0	0	0	0	0
	0	0	0	-1	1	0	0	-1	0	0
	0	0	0	-1	1	0	0	0	-1	1
	0	0	0	0	0	1	-1	0	-1	0
	0	0	0	0	0	1	-1	1	0	-1
	0	0	0	0	0	-1	1	0	-1	0
	0	0	0	0	0	0	0	1	-1	-1
	0	0	0	0	0	-1	-1	0	1	0
	0	0	0	0	0	0	0	-1	-1	1

Figure 3.6 Example of complete  $T$

Such a matrix  $T$  captures the complete per-flow information. For example, at node “a”, it has three adjacent links: links 1, 2 and 3. The 1<sup>st</sup>, 2<sup>nd</sup> and 3<sup>rd</sup> columns of  $T$  record the traffic flows which traverse them and the capacity usage status. Negative values in a given row indicate the links used by the PP, and positive values indicate links used by WP of a given flow. We can further decompose the complete  $T$  into DRM of individual nodes. For example, the local DRM in node a, denoted as  $T\_Node\_a$  only needs to record the capacity usage information related to links 1, 2 and 3 (i.e., highlighted in ‘blue’), see Figure 3.7.

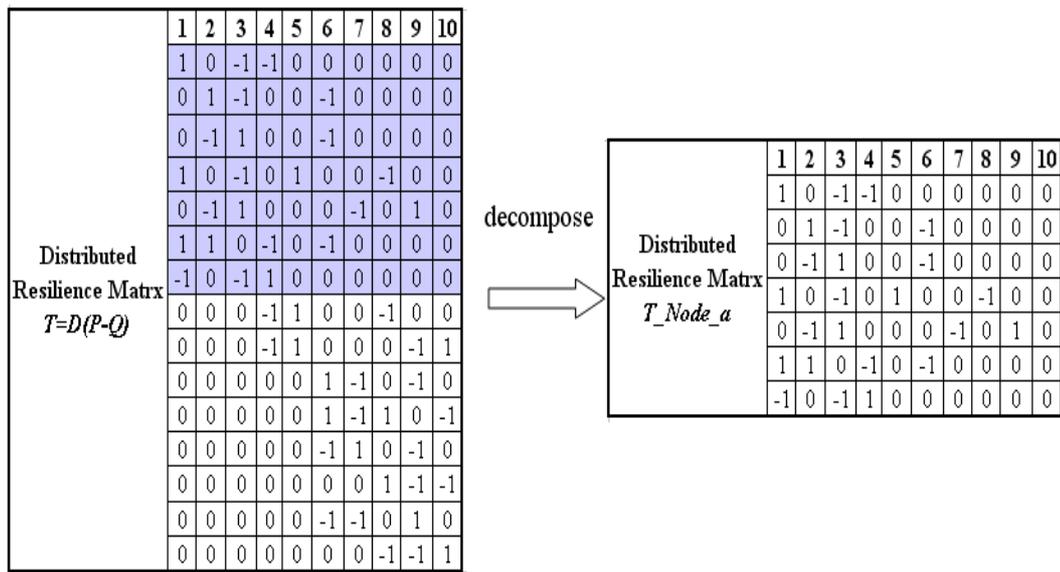


Figure 3.7 The necessary information in  $T$  for links 1-3 at node “a”

Furthermore, the matrix  $T\_Node\_a$  can be further decomposed into three matrices, each for one of its three adjacent links, as seen in Figure 3.8, i.e.,  $T\_Link\_1$ ,  $T\_Link\_2$  and  $T\_Link\_3$ . The mapping rules between each node’s matrix and its adjacent links’ matrix can be specified as follows:

- If flow  $r$  traverses link  $l$  in its working path, then  $T\_Link\_l [n, l] > 0$ , where  $n$  denotes the number of flows that use link  $l$  either in their working path or protection path, and  $n=0; n:=n+1$  as a new flow  $r$  uses link  $l$ ;
- If flow  $r$  traverses the link  $l$  in its protection path, then  $T\_Link\_l [n, m...] < 0$ , where  $m...$  indicates all links traversed in the working path of flow  $r$ , and  $n=0; n:=n+1$  as a new flow  $r$  uses link  $l$ .

For example, flow 1 uses link 1 in its working path, so in  $T\_Link\_1$ :

$$n=0+1 \text{ and } T\_Link\_1 [1, 1] = 1$$

Also, flow 1 uses link 3 in its protection path, so in  $T\_Link\_3$ :

$$n=0+1 \text{ and } T\_Link\_3 [1, 1] = -1$$

In addition, flow 2 uses link 2 in its working path, so in  $T\_Link\_2$ :

$$n=0+1 \text{ and } T\_Link\_2 [1, 2] = 1$$

Flow 2 also uses link 3 in its protection, so in  $T\_Link\_3$ :

$$n=1+1 \text{ and } T\_Link\_3 [2, 2] = -1$$

Following the above mapping, the resulting three  $T\_Link\_l$  matrices for link  $l, l=1, 2$  and 3 adjacent to node “a” is shown in Figure 3.8 below.

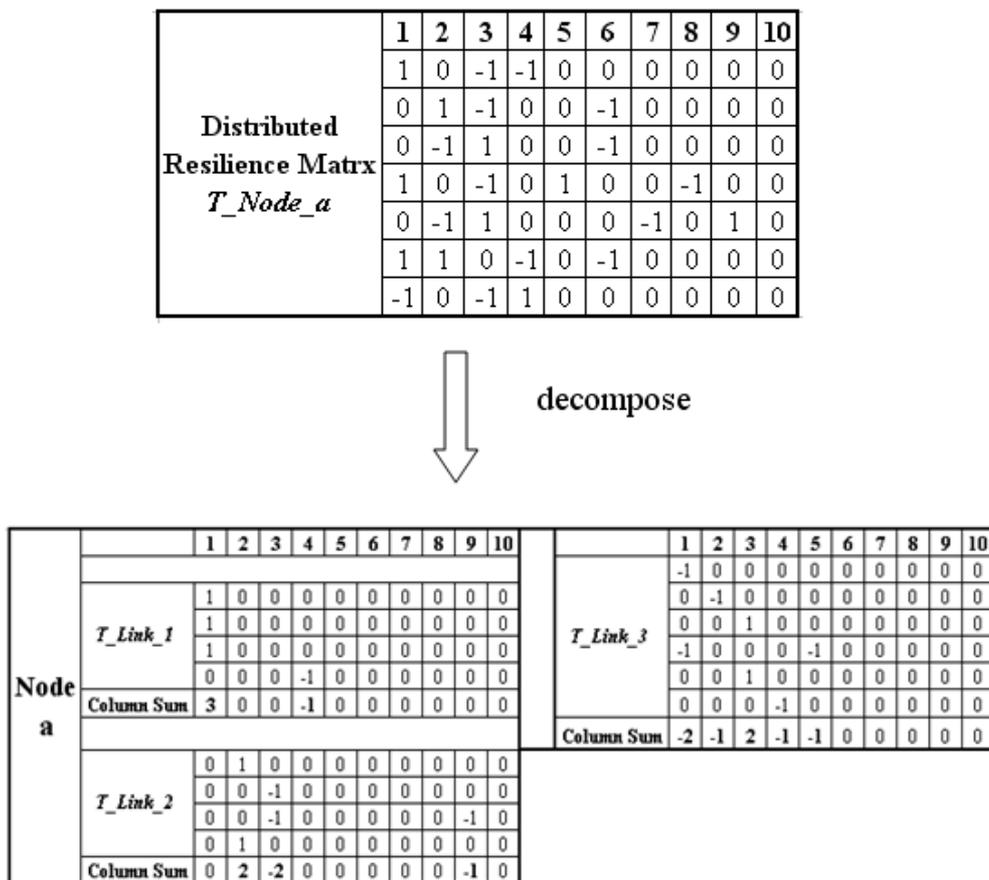


Figure 3.8 Local  $T\_Link\_l, l=1, 2$  and 3 in node a



<b>Node d</b>		1	2	3	4	5	6	7	8	9	10		1	2	3	4	5	6	7	8	9	10		
	<i>T_Link_4</i>		-1	0	0	0	0	0	0	0	0	0	<i>T_Link_8</i>		-1	0	0	0	-1	0	0	0	0	0
			-1	-1	0	0	0	0	0	0	0	0			0	0	0	0	-1	0	0	0	0	0
			0	0	0	1	0	0	0	0	0	0			0	0	0	0	0	0	0	1	0	0
			0	0	0	0	-1	0	0	0	0	0			0	0	0	0	0	0	0	1	0	0
		0	0	0	0	-1	0	0	0	0	-1		0	0	0	0	0	0	0	0	0	0	-1	
	<b>Column Sum</b>	<b>-2</b>	<b>-1</b>	<b>0</b>	<b>1</b>	<b>-2</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>-1</b>	<b>Column Sum</b>	<b>-1</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>-2</b>	<b>0</b>	<b>2</b>	<b>0</b>	<b>-1</b>		
	<i>T_Link_6</i>		0	-1	0	0	0	0	0	0	0	0	<i>T_Link_9</i>		0	0	0	0	0	0	0	0	1	0
			0	0	-1	0	0	0	0	0	0	0			0	0	0	0	-1	0	0	0	0	-1
			-1	-1	0	0	0	0	0	0	0	0			0	0	0	0	0	-1	0	0	0	0
			0	0	0	0	0	1	0	0	0	0			0	0	0	0	0	0	-1	0	0	0
			0	0	0	0	0	1	0	0	0	0			0	0	0	0	0	0	0	-1	0	0
		0	0	0	0	0	0	-1	0	0	0			0	0	0	0	0	0	0	0	1	0	
	0	0	0	0	0	0	0	0	-1	0		0	0	0	0	0	0	0	0	0	-1			
<b>Column Sum</b>	<b>-1</b>	<b>-2</b>	<b>-1</b>	<b>0</b>	<b>0</b>	<b>2</b>	<b>-1</b>	<b>0</b>	<b>-1</b>	<b>0</b>	<b>Column Sum</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>-1</b>	<b>-1</b>	<b>-1</b>	<b>-1</b>	<b>2</b>	<b>-2</b>			
<i>T_Link_3</i>		-1	0	0	0	0	0	0	0	0	0													
		0	-1	0	0	0	0	0	0	0	0													
		0	0	1	0	0	0	0	0	0	0													
		-1	0	0	0	-1	0	0	0	0	0													
		0	0	1	0	0	0	0	0	0	0													
	0	0	0	-1	0	0	0	0	0	0														
<b>Column Sum</b>	<b>-2</b>	<b>-1</b>	<b>2</b>	<b>-1</b>	<b>-1</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>													

Figure 3.11 Local  $T\_Link_l$ ,  $l=3,4,6,8$  and  $9$  in node  $d$

<b>Node e</b>		1	2	3	4	5	6	7	8	9	10		1	2	3	4	5	6	7	8	9	10		
	<i>T_Link_5</i>		0	0	0	0	1	0	0	0	0	0	<i>T_Link_10</i>		0	0	0	0	0	0	0	0	0	1
			0	0	0	0	1	0	0	0	0	0			0	0	0	0	0	-1	0	-1	0	0
			0	0	0	0	1	0	0	0	0	0			0	0	0	0	0	0	0	-1	0	0
			0	0	0	0	1	0	0	0	0	0			0	0	0	0	0	0	0	0	0	1
	<b>Column Sum</b>	<b>3</b>	<b>-1</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>Column Sum</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>-1</b>	<b>0</b>	<b>-2</b>	<b>0</b>	<b>2</b>	
	<i>T_Link_8</i>		-1	0	0	0	-1	0	0	0	0	0												
			0	0	0	0	-1	0	0	0	0	0												
		0	0	0	0	0	0	0	1	0	0													
		0	0	0	0	0	0	0	1	0	0													
		0	0	0	0	0	0	0	0	0	-1													
<b>Column Sum</b>	<b>-1</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>-2</b>	<b>0</b>	<b>0</b>	<b>2</b>	<b>0</b>	<b>-1</b>														

Figure 3.12 Local  $T\_Link_l$ ,  $l=5, 8,$  and  $10$  in node  $e$

		1	2	3	4	5	6	7	8	9	10			1	2	3	4	5	6	7	8	9	10	
<b>Node</b> <b>f</b>																								
	<i>T_Link_7</i>	0	0	-1	0	0	0	0	0	-1	0		0	0	0	0	0	0	0	0	0	1	0	
		0	0	0	0	0	-1	0	0	0	0		0	0	0	0	-1	0	0	0	0	0	-1	
		0	0	0	0	0	-1	0	-1	0	0		0	0	0	0	0	-1	0	0	0	0	0	
		0	0	0	0	0	0	1	0	0	0		0	0	0	0	0	0	-1	0	0	0	0	
		0	0	0	0	0	0	0	0	-1	0		0	0	0	0	0	0	0	-1	0	0	0	
	<b>Column Sum</b>	0	0	-1	0	0	-2	1	-1	-2	0	<b>Column Sum</b>	0	0	0	0	-1	-1	-1	-1	1	-2		
	<i>T_Link_10</i>	0	0	0	0	0	0	0	0	0	1													
		0	0	0	0	0	-1	0	-1	0	0													
	0	0	0	0	0	0	0	-1	0	0														
	0	0	0	0	0	0	0	0	0	1														
<b>Column Sum</b>	0	0	0	0	0	-1	0	-2	0	2														

Figure 3.13 Local  $T\_Link\_l$ ,  $l=7, 9$ , and 10 in node  $f$ 

After all the mappings, the summation of each column in each  $T\_Link\_l$  matrix is performed to obtain a row vector, for example in link 1, we get  $T\_Link\_l[5, \dots] = [3, 0, 0, -1, 0, 0, 0, 0, 0, 0]$ . The negative number in  $T\_Link\_l[5, 4]$  represents the minimum spare capacity required on link 1 when link 4 failed. From this row vector, the capacity usage information for link 1 can be obtained as (3, -1), which means 3 units of working capacity are allocated and at least 1 unit of protection capacity needs to be reserved for any single link failure. This capacity usage information i.e., 1 unit of spare capacity, obtained from our DRM structure is exactly the first value in vector  $s$  if the SPM structure is used. In addition, we can obtain extra information of working capacity usage in the DRM.

The main advantage of the DRM structure is that it can be implemented locally, but still bear the same information on dependencies as that associated with the centralized SPM structure. In addition, although each  $T\_Link\_l$  matrix for link  $l$  is flow-based, but it only needs to record the flow information related to that link, which largely reduces the storage space complexity. For example, there is a total of 15 flows in the prior case, but even the ‘busiest’ link 6 is associated with 7 flows, while the most idle link 5 is only associated with 3 flows. In addition, with the flow-based information, the DRM can efficiently eliminate the bandwidth release ambiguity occurring during traffic demand teardown. For example, as shown in Figure 3.8, if node ‘a’ finishes flow 1 transmission, it can immediately tear down its working path, but, in the general case, deciding about termination of its protection path is problematic. Namely, the source node ‘a’ faces ambiguity on deciding how much capacity

should be released on links 3 and 4. When flow 1 was set up, it had reserved 1 unit of spare capacity which is now also shared by other protection paths. However, if no flow 1- related knowledge is stored at node “a” and “b”, it does not know that other protection paths are sharing this in links 3 and 4 too. In this case, node “a” cannot release the correct amount of capacity without additional knowledge. This limitation results from using only partial network state information for path routing. In the DRM structure, the flow based information is specified in its related  $T\_Link\_l$  matrices, which can avoid the ambiguity problem. In the teardown process, the node “a” only needs to send a ‘flow 1 release’ message to other nodes, and the related nodes can easily delete the spare capacity only contributed by flow 1 in the  $T\_Link\_l$  matrices.

Finally, let us take a closer look at the local  $T\_Link\_l$  matrix of link  $l$  and investigate how the DRM structure allows the distributed computation of capacity usage when a new traffic request  $r_{new}$ , say of 1 unit, is to be sent between nodes “a” and “e”, as shown in Figure 3.14.

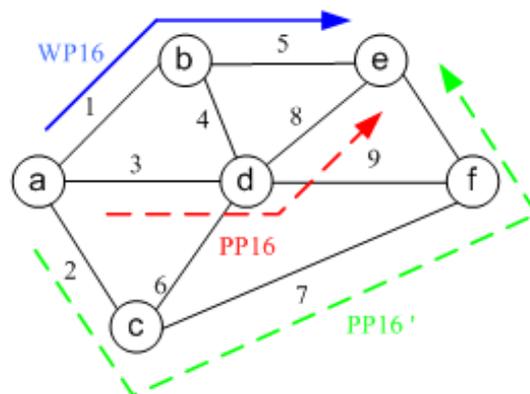


Figure 3.14 The WP and candidate PPs for the new request  $r_{new}$

	1	2	3	4	5	6	7	8	9	10
<i>T_Link_1</i>	1	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0	0
	0	0	0	-1	0	0	0	0	0	0
	1									
Column Sum	4	0	0	-1	0	0	0	0	0	0
<i>T_Link_2</i>	0	1	0	0	0	0	0	0	0	0
	0	0	-1	0	0	0	0	0	0	0
	0	0	-1	0	0	0	0	0	-1	0
	0	1	0	0	0	0	0	0	0	0
	-1				-1					
Column Sum	-1	2	-2	0	-1	0	0	0	-1	0
<i>T_Link_3</i>	-1	0	0	0	0	0	0	0	0	0
	0	-1	0	0	0	0	0	0	0	0
	0	0	1	0	0	0	0	0	0	0
	-1	0	0	0	-1	0	0	0	0	0
	0	0	1	0	0	0	0	0	0	0
	0	0	0	-1	0	0	0	0	0	0
	-1				-1					
Column Sum	-3	-1	2	-1	-2	0	0	0	0	0
<i>T_Link_4</i>	-1	0	0	0	0	0	0	0	0	0
	-1	-1	0	0	0	0	0	0	0	0
	0	0	0	1	0	0	0	0	0	0
	0	0	0	0	-1	0	0	0	0	0
	0	0	0	0	-1	0	0	0	0	-1
Column Sum	-2	-1	0	1	-2	0	0	0	0	-1
<i>T_Link_5</i>	0	0	0	0	1	0	0	0	0	0
	0	0	0	0	1	0	0	0	0	0
	0	0	0	0	1	0	0	0	0	0
					1					
Column Sum	0	0	0	0	4	0	0	0	0	0
<i>T_Link_6</i>	0	0	-1	0	0	0	0	0	-1	0
	0	0	0	0	0	-1	0	0	0	0
	0	0	0	0	0	0	-1	0	-1	0
	0	0	0	0	0	0	0	1	0	0
	0	0	0	0	0	0	0	0	-1	0
Column Sum	0	0	-1	0	0	-2	1	-1	-2	0
<i>T_Link_7</i>	0	-1	0	0	0	0	0	0	0	0
	0	0	-1	0	0	0	0	0	0	0
	-1	-1	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	1	0	0	0
	0	0	0	0	0	0	1	0	0	0
	0	0	0	0	0	0	-1	0	0	0
	0	0	0	0	0	0	0	0	-1	0
	-1				-1					
Column Sum	-2	-2	-1	0	-1	2	-1	0	-1	0
<i>T_Link_8</i>	-1	0	0	0	-1	0	0	0	0	0
	0	0	0	0	-1	0	0	0	0	0
	0	0	0	0	0	0	0	1	0	0
	0	0	0	0	0	0	0	1	0	0
	0	0	0	0	0	0	0	0	0	-1
	-1				-1					
Column Sum	-2	0	0	0	-3	0	0	2	0	-1
<i>T_Link_9</i>	0	0	0	0	0	0	0	0	1	0
	0	0	0	0	-1	0	0	0	0	-1
	0	0	0	0	0	-1	0	0	0	0
	0	0	0	0	0	0	-1	0	0	0
	0	0	0	0	0	0	0	-1	0	0
	0	0	0	0	0	0	0	0	1	0
	0	0	0	0	0	0	0	0	0	-1
Column Sum	0	0	0	0	-1	-1	-1	-1	2	-2
<i>T_Link_10</i>	0	0	0	0	0	0	0	0	0	1
	0	0	0	0	0	-1	0	-1	0	0
	0	0	0	0	0	0	0	-1	0	0
	0	0	0	0	0	0	0	0	0	1
	-1				-1					
Column Sum	-1	0	0	0	-1	-1	0	-2	0	2

Figure 3.15 Updated *T\_Link\_I* for each link according to the new request  $r_{new}$

In Figure 3.14, we have selected  $WP_{16} = (1, 5)$  as the working path, and  $PP_{16} = (3, 8)$  or  $PP_{16}' = (2, 7, 9)$  are under consideration as candidates for its protection path. The *T\_Link\_I* is updated, as depicted in Figure 3.15. All the changed values are highlighted in colours: the changed values of working capacity are highlighted in ‘blue’, and the values highlighted in ‘red’ and ‘green’ are the changed spare capacity when selecting  $PP_{16}$  and  $PP_{16}'$ ,

respectively. The updated information about the working and spare capacity should be sent to the source node by each ingress node of that link. For example with link 8, updated information of (2, -3) , is which interpreted as 2 units of working capacity and 3 unit of spare capacity, is sent to node “a” by node “d”. Based on the capacity usage information collected from all the other nodes, the source node “a” can decide to use PP16’ instead of PP16 as the protection path for WP16, since it is more capacity efficient than PP16 and no extra units of spare capacity are needed if PP16’ is selected: 0 units in link 2, 7 and 9; while 2 extra units are needed if PP16 is chosen: 1 unit on link 3 and 1 unit on link 8. Here, it is only a simple example of how the local capacity usage information is exchanged in DRM structure. In a practical case, this information should be translated into link metrics, which are needed in the routing algorithm to decide the merit of a potential path. In the later Chapter 4, we give more details of how this capacity usage information is used in the FoF-R ant-based routing algorithm.

### 3.3 Extension of DRM for Multiple failures

The DRM structure described above can only handle any single link failure in the network. It can also be generalized to handle multiple failures as the SPM structure does. The novel idea of failure scenario matrix  $F = \{f_k\}_{K \times L} = \{f_{kl}\}_{K \times L}$  was introduced in [53] to improve the SPM to characterize  $K$  failure scenarios. Here,  $L$  is the total number of links and the element  $f_{kl}$  of row vector  $f_k$  in  $F$  equals “1” if link  $l$  fails in scenario  $k$ . In this way, each failure scenario can include a set of links that will fail simultaneously in the scenario. Therefore, the SPM structure is generalized as  $G = Q^T \cdot D \cdot U$ , where  $U = P \odot F^T$ . Note that, the binary matrix multiplication operation “ $\odot$ ” is used here to modify ordinary addition  $1+1=2$  to Boolean addition  $1+1=1$ .

To illustrate how the extension of our DRM structure can handle multiple failures like SPM by borrowing their failure scenario matrix  $F$ , we shall still use the prior example network of six-node, as shown in Figure 3.16. This time, we only assume there are three traffic flows between nodes “a” and “f” with traffic demands of 10, 8 and 12 units, respectively. The corresponding WPs, PPs,  $P$ ,  $Q$ ,  $D$  and  $F$  matrices are illustrated in Figures 3.16 -3.19 as below.

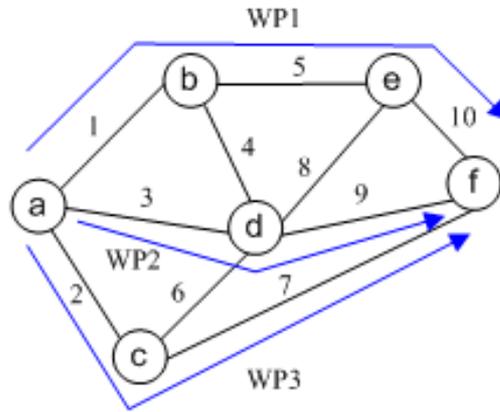


Figure 3.16 Example of working paths

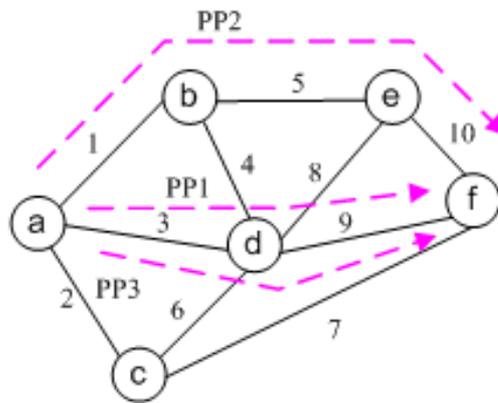


Figure 3.17 Example of protection paths

Flow	Origin	Destination	Working Path (P)										Protection Path (Q)											
			1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10		
1	a	f	1	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0
2	a	f	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	0	1
3	a	f	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0

$$D = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 8 & 0 \\ 0 & 0 & 12 \end{bmatrix}$$

Figure 3.18 Matrices of P, Q and D

$f_{kl}$	1	2	3	4	5	6	7	8	9	10
1	1	0	0	1	1	0	0	0	0	0
2	0	1	0	0	0	1	1	0	0	0
3	0	0	1	1	0	1	0	1	1	0
4	0	0	0	0	1	0	0	1	0	1

Figure 3.19 Failure scenario matrix  $F$ 

As shown in Figure 3.19, the  $F$  matrix covers all the four single node failures i.e., b, c, d and e, excluding the source destination nodes “a” and “f”. Then, we can calculate the SPM  $G$  by matrix operations  $G=Q^T \cdot D \cdot U$  and the resulting  $G$  is shown in Figure 3.20 below.

		Failure Scenario				
		1	2	3	4	s
Links	1	0	0	8	0	8
	2	0	0	0	0	0
	3	10	12	0	10	12
	4	0	0	0	0	0
	5	0	0	8	0	8
	6	0	0	0	0	0
	7	0	0	0	0	0
	8	0	0	0	0	0
	9	10	12	0	10	12
	10	0	0	8	0	8

Figure 3.20 Spare Provision Matrix  $G$  and column vector  $s$ 

The minimum spare capacity required on each link in case of any single node failure in the  $k$  scenario can be obtained by  $s=\max G$ , which  $s$  denotes a vector specifying the maximum element of the corresponding rows of  $G$ .

While in our DRM structure, we first let  $T=D \cdot (P-Q)$ . The complete DRM  $T$  matrix is illustrated in Figure 3.21 below.

		Links									
		1	2	3	4	5	6	7	8	9	10
Flows	1	10	0	-10	0	10	0	0	0	-10	10
	2	-8	0	8	0	-8	0	0	0	8	-8
	3	0	12	-12	0	0	0	12	0	-12	0

Figure 3.21 Example of complete  $T$  in the DRM structure

We can further decompose the complete  $T$  into DRM matrices of individual nodes. For example, the local DRM in node “a”, denoted as  $T_{Node\_a}$  only needs to record the capacity usage information related to links 1, 2 and 3, as seen in Figure 3.22.

		Links									
		1	2	3	4	5	6	7	8	9	10
Flows	1	10	0	-10		10					10
	2	-8	0	8						8	
	3	0	12	-12				12			

Figure 3.22 The necessary information in  $T_{Node\_a}$  for links 1-3

Following the same mapping rules introduced in the previous section, we can finally map the  $T_{Node\_a}$  into three  $T_{link\_l}$  matrices, one for each adjacent link to node “a”, i.e.,  $T_{link\_1}$ ,  $T_{link\_2}$  and  $T_{link\_3}$ .

		Links									
		1	2	3	4	5	6	7	8	9	10
Flows	1	10									
	2	-8		8						8	
	3	0									

↓ mapping

		Links									
		1	2	3	4	5	6	7	8	9	10
$T_{Link\_1}$	1	10	0	0	0	0	0	0	0	0	0
	2	0	0	-8	0	0	0	0	0	-8	0
	3	0	0	0	0	0	0	0	0	0	0
Column Sum		10	0	-8	0	0	0	0	0	-8	0

Figure 3.23 Mapping between  $T_{Node\_a}$  to  $T_{link\_1}$

For example, as shown in Figure 3.23, flow 1 uses link 1 in its working path, so  $T\_link\_1 [1,1]=10$ , while for the flow 2, link 1 is used in its protection path and links 3 and 9 are used in its working path, so  $T\_link\_1 [2, 3] = -8$ ,  $T\_link\_1 [2, 9] = -8$ . The resulting  $T\_link\_l$  matrix for each link  $l$  is shown in Figure 3.24.

	1	2	3	4	5	6	7	8	9	10		1	2	3	4	5	6	7	8	9	10	
$T\_Link\_1$	10	0	0	0	0	0	0	0	0	0	$T\_Link\_6$	0	0	0	0	0	0	0	0	0	0	0
	0	0	-8	0	0	0	0	0	-8	0		0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0
Column Sum	10	0	-8	0	0	0	0	0	-8	0	Column Sum	0	0	0	0	0	0	0	0	0	0	0
$\ominus F^T$	10, 0, -8, 0										$\ominus F^T$	0, 0, 0, 0										
$T\_Link\_2$	0	0	0	0	0	0	0	0	0	0	$T\_Link\_7$	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0
	0	12	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	12	0	0	0
Column Sum	0	12	0	0	0	0	0	0	0	0	Column Sum	0	0	0	0	0	0	0	12	0	0	0
$\ominus F^T$	0, 12, 0, 0										$\ominus F^T$	0, 12, 0, 0										
$T\_Link\_3$	-10	0	0	0	-10	0	0	0	0	-10	$T\_Link\_8$	0	0	0	0	0	0	0	0	0	0	0
	0	0	8	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0
	0	-12	0	0	0	0	-12	0	0	0		0	0	0	0	0	0	0	0	0	0	0
Column Sum	-10	-12	8	0	-10	0	-12	0	0	-10	Column Sum	0	0	0	0	0	0	0	0	0	0	0
$\ominus F^T$	-10, -12, 8, -10										$\ominus F^T$	0, 0, 0, 0										
$T\_Link\_4$	0	0	0	0	0	0	0	0	0	0	$T\_Link\_9$	-10	0	0	0	-10	0	0	0	0	0	-10
	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	8	0	0
	0	0	0	0	0	0	0	0	0	0		0	-12	0	0	0	0	-12	0	0	0	0
Column Sum	0	0	0	0	0	0	0	0	0	0	Column Sum	-10	-12	0	0	-10	0	-12	0	8	-10	0
$\ominus F^T$	0, 0, 0, 0										$\ominus F^T$	-10, -12, 8, -10										
$T\_Link\_5$	0	0	0	0	10	0	0	0	0	0	$T\_Link\_10$	0	0	0	0	0	0	0	0	0	0	10
	0	0	-8	0	0	0	0	0	-8	0		0	0	-8	0	0	0	0	0	-8	0	0
	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0
Column Sum	0	0	-8	0	10	0	0	0	-8	0	Column Sum	0	0	-8	0	0	0	0	0	-8	10	0
$\ominus F^T$	10, 0, -8, 10										$\ominus F^T$	0, 0, -8, 0										

Figure 3.24 Local  $T\_link\_l$ ,  $l=1,2,\dots,10$

After all the mappings, column summation is conducted in each  $T\_link\_l$  to obtain row vectors e.g.,  $T\_link\_1 = [10, 0, -8, 0, 0, 0, 0, 0, -8, 0]$ . The negative numbers here represent the minimum spare capacity required on link 1 when link  $l$  fails. We can see that link 1 is

requires 10 units of working capacity and at least 8 units of protection capacity i.e., “-8” is reserved for any single link failure. Based on this procedure, we now can generalize our DRM structure to handle multiple failures using the failure scenario matrix  $F$ . We also use the Boolean addition  $\odot$  operation such as  $T\_link\_l \odot F^T$  to derive the information on minimum spare capacity required in each link for all single node failure scenarios. For example, in link 1, we can obtain that [10, 0, -8, 0] after performing  $T\_link\_l \odot F^T$ , which informs us that at least 8 units of protection capacity are needed in link 1 when node “d” fails, but any other failure at nodes b, c, e have no impact on link 1. This is exactly the information that can be derived from the 1<sup>st</sup> row of vector  $s$  in  $G$  of SPM in Figure 3.20.

Additionally, we investigate how the  $T\_link\_l$  in DRM structure allows the distributed computation on capacity usage when a new traffic request  $r_{new}$  is arriving between nodes “a” and “f” with 10 unit of capacity requirement. We have selected WP4 = (3, 9) as the working path, PP4 = (1, 5, 10) and PP4' = (2, 7) are under consideration as the candidate for its protection path, as seen in Figure 3.25 below.

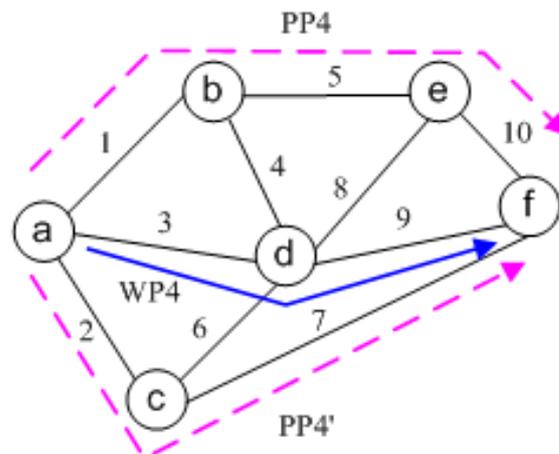


Figure 3.25 The WP and candidate PPs for a new request  $r_{new}$

	1	2	3	4	5	6	7	8	9	10		1	2	3	4	5	6	7	8	9	10	
$T_{Link\_1}$	10	0	0	0	0	0	0	0	0	0	$T_{Link\_6}$	0	0	0	0	0	0	0	0	0	0	
	0	0	-8	0	0	0	0	0	0	-8		0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0
			-10							-10												
Column Sum	10	0	-18	0	0	0	0	0	0	-18	0	Column Sum	0	0	0	0	0	0	0	0	0	0
$\ominus F^T$	10,0,-18,0										$\ominus F^T$	0,0,0,0										
$T_{Link\_2}$	0	0	0	0	0	0	0	0	0	0	$T_{Link\_7}$	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	
	0	12	0	0	0	0	0	0	0	0		0	0	0	0	0	0	12	0	0	0	
			-10							-10											-10	
Column Sum	0	12	-10	0	0	0	0	0	0	-10	0	Column Sum	0	0	-10	0	0	0	12	0	-10	0
$\ominus F^T$	0,12,-10,0										$\ominus F^T$	0,12,-10,0										
$T_{Link\_3}$	-10	0	0	0	-10	0	0	0	0	-10	$T_{Link\_8}$	0	0	0	0	0	0	0	0	0	0	
	0	0	8	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	
	0	-12	0	0	0	0	-12	0	0	0		0	0	0	0	0	0	0	0	0	0	
			10																			
Column Sum	-10	-12	18	0	-10	0	-12	0	0	-10	Column Sum	0	0	0	0	0	0	0	0	0	0	
$\ominus F^T$	-10,-12,18,-10										$\ominus F^T$	0,0,0,0										
$T_{Link\_4}$	0	0	0	0	0	0	0	0	0	0	$T_{Link\_9}$	-10	0	0	0	-10	0	0	0	0	-10	
	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	8	0	0	
	0	0	0	0	0	0	0	0	0	0		0	0	-12	0	0	-12	0	0	0	0	
																				10		
Column Sum	0	0	0	0	0	0	0	0	0	0	Column Sum	-10	-12	0	0	-10	0	-12	0	18	-10	
$\ominus F^T$	0,0,0,0										$\ominus F^T$	-10,-12,18,-10										
$T_{Link\_5}$	0	0	0	0	10	0	0	0	0	0	$T_{Link_{10}}$	0	0	0	0	0	0	0	0	0	10	
	0	0	-8	0	0	0	0	0	-8	0		0	0	-8	0	0	0	0	-8	0	0	
	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	
			-10						-10					-10					-10			
Column Sum	0	0	-18	0	10	0	0	0	-18	0	Column Sum	0	0	-18	0	0	0	0	0	-18	10	
$\ominus F^T$	10,0,-18,10										$\ominus F^T$	0,0,-18,10										

Figure 3.26 Updated  $T_{link\_l}$  for each link according to a new request  $r_{new}$ 

The  $T_{link\_l}$  is updated, as seen in Figure 3.26. Here, all the changed values are highlighted in colour, where ‘red’ is used for selecting PP4 and ‘blue’ is used for selecting PP4’. After the local re-calculation, it can be seen that, using the PP4’ is more efficient than PP4, since only 20 extra units of capacity are needed: 10 units in link 2 and 10 units in link 7, while 30 units are needed if PP4 is chosen: 10 units each on links 1, 5 and 10. Therefore, it shows that the DRM structure work well in its extensions to deal with multiple failure scenarios.

In conclusion, it can be seen that, the capacity usage in each link and the information about the relationship between working and protection capacity can be well captured by the DRM framework in a distributed manner. We highlight that, in the DRM, the essential information on the links traversed by the working path for a traffic demand needs to be

known by its protection links. Therefore, for a traffic demand between specific source and destination nodes, a route searching message needs to memorize the links traversed in its working path, and then delivers this information to update the local  $T\_Link\_I$  matrix entries of each link in its corresponding protection path as well as uploading the updated capacity usage information for exchange. This sort of distributed signaling needs to be implemented by a special distributed control mechanism. In the next chapter, we propose a Friend-or-Foe Resilient (FoF-R) ant-based routing algorithm to implement it and also can jointly optimize the working and protection paths routing problem. We shall describe how the heuristic FoF-R ant algorithm with the DRM structure can find the optimal protection cycles and also explore the capacity sharing potential among protection paths in the network.

### **3.4 Summary**

In this chapter, we developed a novel and comprehensive distributed framework, called DRM, to capture the dependency between the working and protection capacity. We have specified the information to be maintained as well as how it is updated and exchanged through distributed signaling. We highlighted that, in the DRM structure, the essential information on the links traversed by the corresponding working path needs to be known to the links in its protection path. This information exchange needs to be implemented by a special distributed signaling mechanism.



# Chapter 4

## FoF-R Ant-based Routing

### Algorithm

Telecommunication networks should be designed to be robust and to meet the capacity allocation related resilience requirements, so as to provide uninterrupted communication services in the case of failures. Moreover, providing high resiliency and maintaining QoS attributes is very costly in terms of the extra required transmission, switching and routing facilities. Hence, it is critical that the resiliency (or hence cost) of a network, be tailored to meet strict budget requirements. In this chapter, our objective is to develop a survivable routing algorithm to satisfy the requirements of network robustness, adaptability, and distributiveness while implementing the Distributed Resilience Matrix (DRM) framework introduced in Chapter 3 by an ant-based algorithm. This kind of heuristic approach is commonly known as emergent systems, swarm intelligence, or biologically inspired systems. The ant-based heuristic is specially introduced here to tackle the constrained multi-commodity, multi-criteria SCA optimization problem for survivable routing. Different from the traditional ant algorithm, our novel Friend-or-Foe Resilient (FoF-R) ant-based routing algorithm with DRM framework is proposed to find the optimal protection cycle (i.e., two node disjoint paths between a source-destination node pair) and exploring the capacity sharing ability among protection paths using a capacity headroom-dependent attraction and repulsion function.

The remainder of this chapter is organized as below. First, we introduce the basics of swarm intelligence and ant colony optimization (ACO). The previous successful ant-based algorithms for routing and load balancing e.g., AntNet and Multiple Ant Colony

Optimizations (MACO) are also reviewed. The FoF-R ant-based routing algorithm is presented. Simulation results based on the OMNeT++ tool show that the FoF-R scheme with the DRM is a promising approach to solve the SCA problem for survivable routing and it gives a good tradeoff between a solution's optimality and the time needed for finding a solution.

#### 4.1 Introduction on Ant Colony Optimization

Research in ethnology suggests that self-organization is an important component of many collective phenomena in a society of insects [61-65]. The theory of self-organization was developed originally in the context of physics and chemistry, and can be extended to social insects to show that complex collective behaviour may emerge from interactions among individuals that exhibit simple behavior. A swarm intelligence system defined in [66] is based on the algorithms or distributed problem-solving devices inspired by the collective behaviour of social insect colonies or other animal societies. It is generally understood to be the result of overall behaviour generated by many simple behaviors interacting in some way. Emergent behaviour is not easily deductible from a description of the simple behaviors generating it. For example, ant-colonies, a typical social insect society, can be considered as a distributed system and in spite of the simplicity of each individual, ant-colonies present a highly structured self-organization. Ant-colonies can accomplish complex tasks that in some cases far exceed the individual capacity of a single ant. An illustration of the collective foraging behaviour of many ant species is the finding shortest path experiment [64, 67] as shown in Figure 4.1 below.

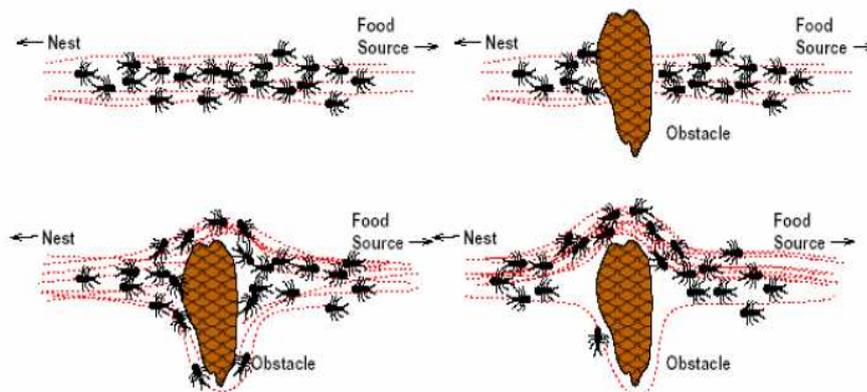


Figure 4.1 Finding the shortest path around an obstacle

Figure 4.1 illustrates how ants can reroute a trail around an obstacle which suddenly appears and find a shorter path to the food source. In this experiment, a food source is separated from the nest by a sudden obstacle (i.e., a cone). It is observed that in most experiments, the shorter path (up one) is selected by the colony if the down path is sufficiently longer. This is because these ants have a trail-following behavior: an individual ant lays a chemical substance, called a pheromone, which attracts other ants. The ants which took the shorter path (i.e., up one) returning to the nest will be faster, and influence the outgoing ants toward that shorter path, which becomes marked more and more strong. Therefore, this ant's foraging behaviour is able to find the optimal shortest path. Four fundamental processes operated in this biological ant-based system are:

- **A large number of ants** try simultaneously and asynchronously to move themselves from the nest to the food source;
- **Communication is indirect** between the ants, i.e. they leave chemical trails of messages i.e., pheromones on the ground. All ants can smell each others' pheromone;
- **A stochastic process** can be used to model the navigational behaviour of an ant. By combining information about the neighbourhood, e.g. placement of a nearby obstacle like the cone, and the distribution of pheromone on the ground, ant movement is ruled by a probability distribution. A selection from the distribution produces a directional vector, which controls the ant's forward movement. The higher pheromone intensities associate with higher probabilities of selection on specific path, thus the likelihood of following a trail used by many ants is higher than exploring a new route;
- **The search is iterative.** Ants move along the same or similar trail again and again, which are continuously updating the pheromone levels on the ground.

From the ant foraging experiment described above, we can extract three main properties, which are relevant to the network resiliency context and thus inspire us to develop the new FoF-R ant-based survivable routing algorithm:

- **Adaptiveness:** The simple behaviour typically handles unexpected responses from interactions with the environment. Such mechanisms often include a stochastic

component, i.e. by a random choice, a response is decided in one of a set of possible ways;

- **Robustness:** Weak inter-component dependencies reduce the probability of system breakdown due to individual component failures. This is ensured by lack of synchronized hierarchical control and use of asynchronous indirect communication by changing the environment. Redundancy further reduces the probability of system breakdown due to individual component failures.
- **Efficiency:** When encountered with complex problems, e.g. problems of the NP classes, emergent systems tend to find near optimal solutions with great efficiency. Reasons for this efficiency can be traced to the interplay of positive and negative feedback mechanisms and the stochastic (adaptive) mechanisms mentioned.

#### 4.1.1 Ant Colony Optimization

Inspired by intelligence obtained from the ant colony's foraging behavior, the Ant Colony Optimization (ACO), a new meta-heuristic for optimization has been proposed in [68-69]. This meta-heuristic has been applied to classical optimization problems, such as the traveling salesman problem [70], the quadratic assignment problem [71] and the job-shop scheduling problem with great success. This method, as a general heuristic, can be compared with simulated annealing [72]. Some ant-based optimization methods extended from the original method have been applied to the vehicle routing problem [73], graph colouring problem [74] and search of continuous spaces [75].

In ACO algorithms, a finite size colony of artificial ants collectively searches for good quality solutions to the optimization problem. Each ant builds a solution, or a component of it, starting from an initial state selected according to the problem dependent criteria. When building its own solution, each ant collects information on the problem characteristics and on its own performance, and uses this information to modify the representation of the problem, as seen by the other ants. Ants act concurrently and use indirect communication. An incremental constructive approach is used by the ants to search for a feasible solution. A solution is expressed as a minimum cost (or shortest) path through the states of the problem in accordance with the problem's constraints. The complexity of each ant is such that even a single ant is able to find a (probably poor quality) solution. High quality solutions are only found as the emergent result of the global cooperation among all the agents of the colony concurrently building different solutions. Each ant builds a solution by moving through a

(finite) sequence of neighbour states. Movements are selected by applying stochastic local search policy directed by:

- ant individual information (the ant internal state, or memory);
- publicly available pheromone trail
- a priori problem-specific local information.

Moreover, the releasing of pheromone depends on the characteristics of the problem. Ants can release pheromone while building the solution, or after a solution has been built, moving back to all the visited states. The autocatalysis plays an important role in ACO algorithms: the more ants choose a move, the more the move is rewarded by adding pheromone and thus more attractive it becomes for the next ant. In general, the amount of pheromone deposited is made proportional to the goodness of the solution an ant has built or is building. In this way, if a move contributed to generate a high-quality solution, its goodness will be increased proportionally to its contribution. A functional composition of the locally available pheromone and heuristic values define ant decision tables, that is, probabilistic tables used by the ants' decision policy to direct their search towards the most attractive regions of the search space. The stochastic component of the movement choice decision policy and the pheromone evaporation mechanism avoid a rapid drift of all the ants toward the same part of the search space. Once an ant has accomplished its task, consisting of building a solution and depositing pheromone information, the ant dies and it will be deleted from the system.

The ACO algorithms, as a consequence of their concurrent and adaptive nature, are particularly suitable for distributed stochastic problems where the presence of exogenous sources determines a non-stationary in the problem representation in terms of costs and/or environment. More details about the ACO meta-heuristic, as well as of the class of problems to which it can be applied, can be found in [76].

#### **4.1.2 ACO Algorithms for Network Routing**

Research shows that current network routing algorithms are not adequate to tackle the increasing complexity of modern wide-area networks [77]. Centralized algorithms have scalability problems; static algorithms have trouble on keeping up-to-date with network changes; and other distributed and dynamic algorithms have oscillations and stability problems. Mobile agents are a promising technique for network routing and management

[78-79] as well as a novel way of building distributed software systems [80]. Unlike traditional stationary algorithms, mobile agents are small packets that can move themselves from node to node, cooperate with others to perform complex tasks in a distributed manner. These agents explore and collaborate in a network, collect routing information and update nodes' routing tables, such as a routing path can be determined for data transmission.

A number of ant-based routing algorithms have been proposed. The most celebrated one is AntNet [81, 90], an adaptive agent-based routing algorithm that has outperformed the best-known routing algorithms on several packet-switched communication networks. It was given another interesting example using a variation of swarm routing based on Bellman's principle of dynamic routing in [82]. In addition, there are a number of ant-based routing algorithms for mobile and ad hoc networks e.g., in [83-88]. Extensive results of ant-based routing in communication networks have shown that this approach is very flexible and can achieve good performance in comparison with conventional routing methods.

In [89] Schoonderwoerd et al have developed an ant-based system for call routing in telecommunication networks which mimics, to some degree, the biological systems described above. Ants are implemented as mobile agents and pheromones as probabilities in matrices. A separate matrix is generated for each relevant source and destination pair, e.g. the matrix  $p_i^{od}$  in Figure 4.2 represents pheromones "pointing" from the nest "o" towards the food source "d."

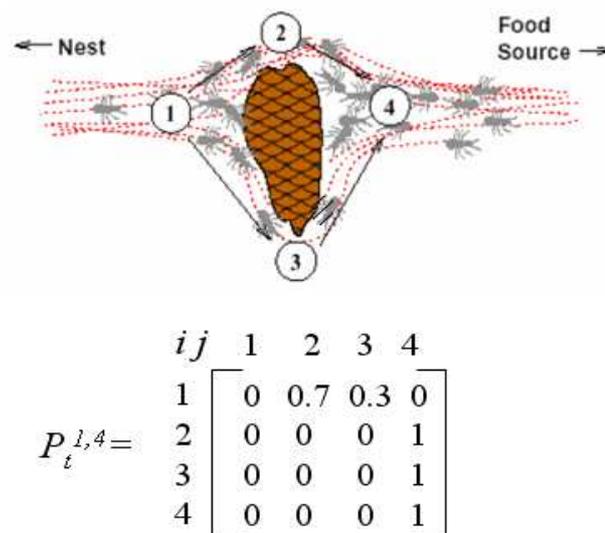


Figure 4.2 Probability matrix at time  $t$  from the obstacle scenario

The probability matrix is relevant for an agent moving from the nest  $o=1$  towards the food source  $d=4$  at time  $t$ , which is applying the resulting probability matrices as the routing table for packets (or connections) routing in a network. Symmetric traffic conditions are assumed, i.e. delays experienced by agents moving forward must be representative for the traffic flowing in the opposite direction.

The AntNet [90] has been introduced to handle asymmetric traffic conditions. In this new system, an ant-like agent performs a search for a path in two phases. During the first phase, the forward phase, the agent only reads probabilities (i.e., smells pheromones) and navigates according to the processes described above. When reaching the destination node, the path quality is calculated and a reinforcement value  $r$  estimated. During the second phase, agent backtracks along the path found to the source node. When moving from node  $j$  to node  $i$ , the probability  $p_{t,ij}^{od}$  of matrix  $p_t^{od}$  pointing forwards from source node “ $o$ ” towards destination node “ $d$ ” of the path is updated in the same manner as described above. They also show that their AntNet system can produce routing tables of high quality. When packets in a packet switched network are routed stochastically using the probability tables generated by AntNet, efficient load balancing and low delays are achievable. Fast route reconfiguration is also provided when network topology changes occur.

In the AntNet algorithm, routing is determined by means of interactions of forward and backward network exploration agents (i.e., ants). The idea of using two-way agents is that the backward ants utilize the useful information gathered by the forward ants on their trip from source to destination. Based on this principle, no node routing updates are performed by the forward ants. Their only purpose in life is to report network delay conditions to the backward ants, in the form of trip times between each network node. The backward ants inherit this raw data and use it to update the routing table of the nodes. The entries of the routing table are probabilities, and as such, must sum to 1 for each row of the network. These probabilities serve a dual purpose:

- the exploration agents of the network use them to decide the next hop to a destination, randomly selecting among all candidates based on the routing table probabilities for a specific destination;
- the data packets deterministically select the path with the highest probability for the next hop.

The AntNet algorithm works as follows:

- Each network node launches forward ants to all destinations in regular time intervals;
- The ant finds a path to the destination randomly based on the current routing tables;
- The forward ant creates a stack, pushing in trip times for every node as that node is reached;
- When the destination is reached, the backward ant inherits the stack;
- The backward ant pops the stack entries and follows the path in reverse;
- The node tables of each visited node are updated based on the trip times.

All of the ant-based algorithm introduced above have addressed the problem of routing but not load balancing. Only one probabilistic routing table is maintained in each node. Consequently, if there is more than one optimal path, then it will be more likely for all data traffics to be directed into only one of the optimal paths. One of the possible solutions is to maintain multiple probabilistic routing tables in a node. An ongoing work has addressed this issue under the topic of multiple ant colony optimizations [91-94]. In these studies, more than one colony of ants is used to search for optimal paths, and each colony of ants deposits a different type of pheromone represented by a different colour. Although ants in each colony respond to pheromone from its own colony, it is augmented with a repulsion mechanism that prevents ants associated with different colonies from choosing the same optimal path.

Some research work in [93] has adopted ACO to solve problems in virtual wavelength path routing and wavelength allocations. The distinguishing feature of the three variants of their ACO algorithms is that ants are not only attracted by the pheromones of other ants in their own colonies, but they are also repelled by the pheromones of other colonies. The motivation of their work stems from the fact that virtual wavelength paths can only carry a limited number of different wavelengths because of technological limitations and cost implications. In the virtual wavelength path routing, the problem is to allocate the minimum number of wavelengths for each link by evenly distributing the wavelength requirements over different links, while at the same time keeping the path lengths short (e.g., in terms of hop numbers). While pheromone attraction is used in the similar sense as other routing applications of ACO, pheromone repulsion enhances the chance of distributing different wavelengths over different links.

## 4.2 FoF-R Ant-based Survivable Routing Algorithm

As mentioned above, the ant-based routing algorithms developed so far have concentrated on the coordination behaviour between agents, and there has been little work put on exploring it for the SCA problem for survivable routing. Competitive behaviour for disjoint path finding was developed in [95-96]. The former introduces collaboration and competition strategy, known as Multiple Ants Colony Optimization System (MACO), by using ants with different types of pheromones: ants cooperate with others if pheromone is of the same type as their own, or they compete if pheromone is of a different type. In [96], a distributed routing algorithm based on the cross-entropy method is proposed. These studies only concentrated on disjoint path finding, and do not target the expose of spare capacity allocation optimization for survivable routing.

Inspired by these prior works, the shared path protection scheme in survivable routing could benefit significantly from a certain level of repulsive behaviour between agents. Therefore, we propose the Friend-or-Foe Resilient (FoF-R) ant-based routing algorithm, which is armed with an attraction and repulsion relationship function. Compared with the traditional ant algorithm, the FoF-R ant algorithm requires an ant agent that has an ability to identify the relationship to a previously laid pheromone, so as to decide what action to take: “friend” (i.e., attraction) or “foe” (i.e., repulsion). If “friend” is recognized, the idea is vivid: friends are attractive to each other to traverse the same protection path (PP) to maximize the capacity sharing. On the other hand, the recognition of “foe” makes the ants detest each other, which means they need to follow disjoint routes and thereby avoid overloading the protection path. The capacity headroom-dependent function (CHF) is introduced to model the Friend and Foe relationships.

### 4.2.1 FoF-R Ant Agent

FoF-R ant agent is proposed to discover a set of node-disjoint cycles between every node pair in the network, each cycle is composed of two node-disjoint paths. These cycles are always available and updated before the connection request arrives, thus we can easily select the best cycle among them for connection setup. This method can guarantee a small setup delay because these cycles already exist by the time of connection setup. Following the ant-colony routing principle, the mobile agents can report to each network node the cycles with low cost to increase the network performance.

In the FoF-R ant-based algorithm, the routing is determined by means of interactions of network exploration agents i.e., ants. The core idea of using ant agents to find a cycle which is formed by the node-disjoint working and protection paths is that the ants can deliver the essential information e.g., the link traversed and capacity usage in its working path from source to destination, to the links in its protection path from destination to source. This is quite different from the previous AntNet algorithm, in which the backward ant is actually tracking back to its working path only. Based on this finding cycle principle, the updating on the local  $T\_link\_l$  matrix for link  $l$  at each node and the exchanging of the essential capacity usage information between nodes in the DRM structure described in Chapter 3, can be implemented through these smart FoF-R moving ants.

#### 4.2.2 Capacity Headroom-dependent Function

Let a traffic flow  $r$  be specified by its source-destination node pair  $(o, d)$  with capacity requirement  $d_r$ . Let  $W_l$  denote the set of working paths that use link  $l$  and  $P_l$  is the set of protection paths that use link  $l$ . Let  $C_l$  denotes the total capacity of link  $l$ ,  $C_l^W$  is the total working capacity used and  $C_l^P$  is the spare capacity reserved in link  $l$ . Therefore, we have:

$$C_l^W = \sum_{r \in W_l} d_r \text{ and } C_l^P = \max\{d_r\}, r \in P_l, \quad (4.1)$$

Where the values of  $C_l^W$  and  $C_l^P$  can be easily obtained from the column summation in  $T\_link\_l$  matrix for link  $l$  in DRM structure. Thus, the capacity headroom of link  $l$ , denoted as  $h_l$ , which is the ratio of the residual capacity to the total capacity in link  $l$ , can be calculated as:

$$h_l = 1 - \frac{(C_l^W + C_l^P)}{C_l}, h_l \in [0, 1] \quad (4.2)$$

Therefore, the Capacity Headroom-dependent Function (CHF), denoted as  $R_l(h_l)$ , is introduced to model the Friend and Foe relationships can be calculated as below.

$$R_l(h_l) = -a \cdot h_l + b \cdot \frac{1}{h_l^2} \quad (4.3)$$

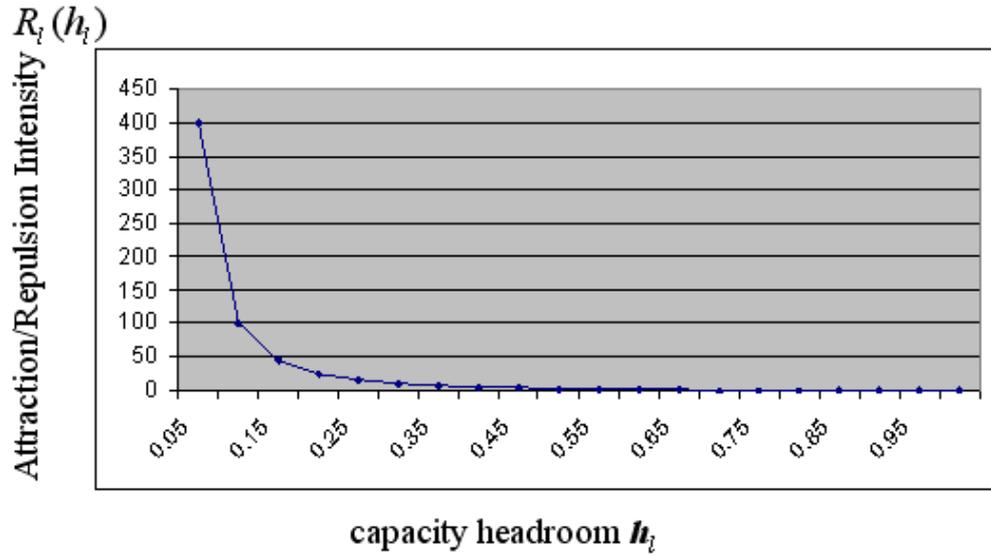


Figure 4.3 An example of  $R_i(h_i)$

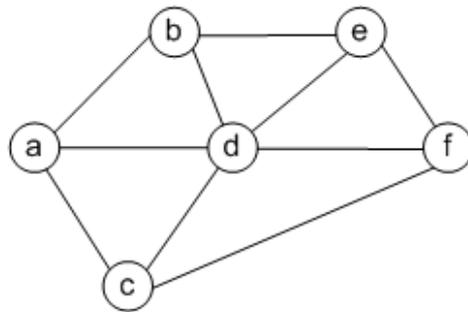
Figure 4.3 shows an example curve of CHF,  $R_i(h_i)$ , where  $a = b = 1$ . The parameters  $a$ ,  $b$  are positive constants, with  $a$  representing the degree of attraction and  $b$  the degree of repulsion. The  $R_i(h_i)$  can model a trust degree from unbounded repulsion (i.e., indefinite link cost) as Foe relationship to a linear attraction (i.e., small link cost) as Friend relationship when the capacity headroom  $h_i$  increases. In other words, the  $R_i(h_i)$  is attractive i.e.,  $-a \cdot h_i$  dominates for large  $h_i$ , or repulsive (i.e.,  $b/h_i^2$  dominates) for small  $h_i$ , which is consistent with inter-individual attraction and repulsion phenomenon in biological swarms. We can adjust the  $(a, b)$  pair to specify the different trust degree in this relationship function. In addition, the  $R_i(h_i)$  is used as the heuristic information contributed to the next-hop selection probability in routing table, which is introduced next.

#### 4.2.3 FoF-R Ant Routing Table Structure

To support the dynamic route selection, in FoF-R ant-based algorithm, a node  $i$  with  $\mathcal{N}_i$  neighbours has a routing table  $\mathcal{T}_i = [\tau_{ij}]_{N-1, j}$ ,  $j \in \mathcal{N}_i$  with  $N-1$  rows and  $|\mathcal{N}_i|$  columns. Here  $N$  is the total number of network nodes,  $\mathcal{N}_i$  is the set of all the neighbour nodes of node  $i$  and  $|\mathcal{N}_i|$  is the number of neighbour node of node  $i$ . Each row corresponds to a

destination node and each column corresponds to a neighbour node. The value  $\tau_{ijd}$  expresses the probability of selecting neighbour node  $j$  as the next hop when an ant moves toward its destination node  $d$  from node  $i$ . For each destination, the sum of all neighbours' selection probabilities must be 1 to satisfy the normalized condition:

$$\sum_{j \in N_i} \tau_{ijd} = 1, \quad d \in [1, N-1] \quad (4.4)$$



Destination	Neighbor		
	b	d	c
b	0.7	0.2	0.1
c	0.1	0.2	0.7
d	0.1	0.8	0.1
e	0.7	0.2	0.1
f	0.1	0.7	0.2

Figure 4.4 An example of routing table at node “a” in a simple network with 6 nodes

An example of the routing table is shown in Figure 4.4. When a connection request occurs between source node “a” and destination node “f”, according to the selection probability in the routing table, node “d” will be selected as the next hop because  $\tau_{a,d,f} > \tau_{a,c,f} > \tau_{a,b,f}$ .

#### 4.2.4 FoF-R Ant Routing Table Updating

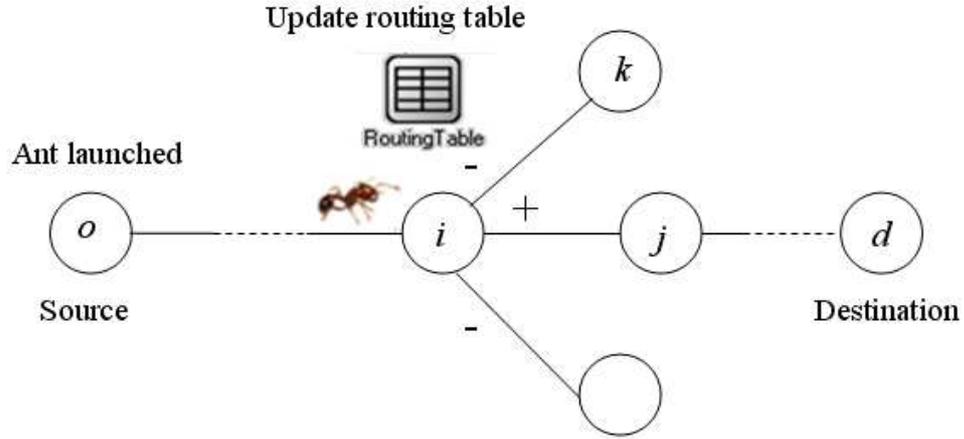


Figure 4.5 FoF-R ant's routing table updating

As shown in Figure 4.5, when an ant visits a node, it updates the entries in the routing table. For example, an ant moves from the source node “o” to destination node “d” following the route  $(o, \dots, i, j, \dots, d)$ , it updates the entries corresponding to the source node “o” in the routing table of node  $i$  as follows: the probability of selecting neighbour  $j$  is increased while the probabilities of selecting other neighbours are decreased. We assume that an ant visits node  $i$  at time  $t$ , so the values for routing entries in time  $t+1$  in node  $i$  are determined as the following :

$$\tau_{i,j,d}(t+1) = \frac{\tau_{i,j,d}(t) + \delta_r}{1 + \delta_r} \quad (4.5)$$

For the other nodes that is different from the visited node  $j$ :

$$\tau_{i,k,d}(t+1) = \frac{\tau_{i,k,d}(t)}{1 + \delta_r}, \forall k \in \mathcal{N}_i, k \neq j \quad (4.6)$$

Here,  $\delta_r$  is the reinforcement parameter and is derived from the data collected by the ant. In FoF-R ant-based routing algorithm, this parameter is calculated as:

$$\delta_r = \alpha \cdot \delta_p + (1 - \alpha) \cdot \delta_c \quad 0 \leq \alpha \leq 1 \quad (4.7)$$

where  $\delta_p$  is the amount of pheromone corresponding to the path length and  $\delta_c$  is the amount of pheromone corresponding to the capacity usage information. In addition, we introduce a scalar parameter  $\alpha$  such that we can adjust the weight between  $\delta_p$  and  $\delta_c$  in  $\delta_r$ .

The factor  $\delta_p$  is derived from the length of the path that the ant has moved along. The shorter the path length, the bigger the  $\delta_p$  value is, and vice versa. Note that the length  $p$  of a path between a source-destination node pair is always greater than or equal to the length of the shortest path between the source and the destination, denoted by  $p_{\min}$  hereafter, and as a reinforcement parameter,  $\delta_p$  must be small as  $0 < \delta_p < 1$ . Thus, we compute  $\delta_p$  as follows:

$$\delta_p = e^{-\beta \Delta p}, \quad \Delta p = p - p_{\min} \quad (4.8)$$

where  $\beta$  is a control parameter. Since the absolute value of path length varies significantly in a large network, using the length difference instead of the absolute length in determining the reinforcement parameter  $\delta_p$  enables the pheromone updating process for all node pairs to be controlled by the same parameter  $\beta$ . The factor  $\delta_c$  is corresponding to the capacity usage information. The link with more free capacity has a larger value of  $\delta_c$ . Similar to  $\delta_p$ ,  $\delta_c$  should be a small value and  $0 < \delta_c < 1$ , we can calculate  $\delta_c$  based on the CHF  $R_l(h_l)$  as described previously:

$$\delta_c = e^{\gamma R_l(h_l)} - 1 \dots \dots \dots (4.9)$$

where  $\gamma$  is another control parameter. Here  $\alpha$ ,  $\beta$  and  $\gamma$  are design parameters and can be adjusted to tune the system performance. The value of  $\alpha$  weighs the importance of the capacity usage value with respect to the path length value. An ant's decisions are therefore taken on the basis of a combination of a long-term learning process i.e., path length and an instantaneous heuristic i.e., link capacity usage.

Moreover, as an ant moves from a source to a destination, its next hop is determined stochastically: a neighbour is selected according to its selection probabilities in the routing table. This is the basic principle of ant colony optimization followed by our FoF-R ant

algorithm. As a result, FoF-R ant tends to discover the better path between a node pair in terms of path length and capacity usage along this path. In our algorithm, an ant is killed when it reaches its source node after a cycle finding procedure. An ant is also killed if its lifetime exceeds a predefined value TTL (Time-To-Live). Ant-based algorithms usually suffer from stagnation, in which an optimal path is found by ants so the pheromone for this path is recursively increased. In this case, too many ants concentrate on this optimal path, which prevents them from discovering other better paths when the network state changes. To avoid this “local optima” situation, a random exploration factor  $P_{noise}$  is introduced in our FoF-R algorithm: at each node; the FoF-R ant selects its next hop randomly with an exploiting probability  $P_{noise}$  and selects its next hop according to the routing table with probability  $1 - P_{noise}$ . The using of  $P_{noise}$  allows ants to keep exploring for a better solution for a traffic request.

With support from the routing table and the ant’s foraging, path selection can be performed in a straightforward manner as described in all other ant-based algorithms: when a connection request arrives at the source node, the next hop will be determined by the node with the highest selection probability or a probability of  $P_{noise}$  to choose a random node among all its neighbouring entries.

#### 4.2.5 FoF-R Routing Procedure

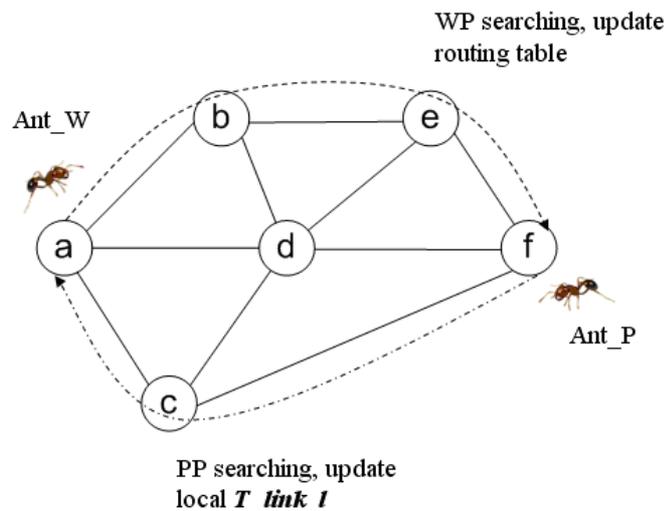


Figure 4.6 FoF-R ant-based routing procedure

The main behaviour of FoF-R ant agent is illustrated in Figure 4.6 above. Here we use Ant\_W to denote FoF-R ant with working path search status, and Ant\_P as FoF-R ant with protection path search status. Each FoF-R ant needs to carry a memory stack  $\mathcal{M}_{o \rightarrow d \rightarrow o}(i)$  of data, where  $i$  refer to the  $i$ th visited node in its cyclic journey. Let  $\mathcal{N}_i$  be set of neighbouring nodes outgoing from node  $i$  and  $\tau_{ijd}$  be the probability that the ant jumps from node  $i$  to node  $j$ ,  $j \in \mathcal{N}_i$ . The FoF-R ant-based distributed routing procedure can be described as follows:

**START**

{

**Initialization of Routing table and  $T\_link\_l$ :** for each node  $i$ , the routing table is initialized with a uniform distribution and its  $T\_link\_l$  for each adjacent link  $l$  is set as a  $1 \times L$  matrix such as:

$$\tau_{ijd} = \frac{1}{|\mathcal{N}_i|}, \forall j \in \mathcal{N}_i \text{ and } T\_link\_l [1][L] = \{0\}$$

**DO** (in parallel)

{

**STEP 1:** In regular time intervals, each node “ $o$ ” launches a FoF-R ant to a randomly chosen destination “ $d$ ”, its mission status is set as Ant\_W

**DO** (in parallel, as each Ant\_W)

{

**STEP 2:** After visiting a node  $j$ , Ant\_W pushes in its stack  $\mathcal{M}_{o \rightarrow d \rightarrow o}(j)$  the information about the previously traversed link  $l$  and the time between its launching to its arriving at  $j$ . Ant\_W selects the next node to visit in the following way:

It decides a movement based on the routing table  $\mathcal{T}_i = [\tau_{ijd}]_{N-1, j}$ ,  $j \in \mathcal{N}_i$ ,

or with a probability of  $P_{noise}$  to choose a random node among its neighbours

**IF** A loop is found

FoF-R ant pops from its stack all data for the loop nodes to avoid infinite loops

**END IF**

} **WHILE** jumping node  $j \neq d$

**STEP 3:** When arrives at the destination  $d$ , FoF-R ant changes its mission status to Ant\_P

**DO** (in parallel, as each Ant\_P)

{

In its return journey, FoF-R ant delivers the traversed links' information into  $T\_link\_l$  in its protection path. We can obtain  $R_l(h_l)$  to calculate  $\delta_c$  and then  $\delta_r$  to update routing table and also according to  $\tau_{ijd}$  to select the next hop  $j$

} **WHILE** ( $j \neq o$ )

**STEP 4:** The source node evaluates the goodness associated with each protection cycle found by the FoF-R ant, by using the following additive link cost function:

$$L(o, d) = \sum_{l \in cycle(o, d)} R_l(h_l) \quad (4.10)$$

**STEP 5:** Update the routing table of each node visited in the best cycle, i.e., increasing the pheromone probabilities of links that cycle used and decrementing, by normalization, the other links' pheromones.

**STEP 6:** All the good protection cycles stored at the source node, are constantly updated, thus the source node can easily select the best quality cycle for each connection setup.

} **END**

### 4.3 Implementation of FoF-R on OMNeT++

#### 4.3.1 Introduction to OMNeT++

OMNeT++ stands for Objective Modular Network Testbed in C++ in [97]. It is a discrete event simulation tool designed to simulate computer networks, multi-processors and other distributed systems associated with GUI simulation library debugging and tracing. Its applications can be extended to modeling other systems as well. It has become a popular network simulation tool in the scientific community as well as in industry over the years. A model network consists of “nodes” connected by “links”. The nodes representing blocks, entities, modules, etc, while the link representing channels, connections, etc. The structure of how fixed elements (i.e. nodes) in a network are interconnected together is called the topology. OMNeT++ uses the NED (Network Description) language, thus allowing for a more user friendly and accessible environment for creation and editing. It has a human-readable textual topology and also uses the same format as that of a graphical editor. OMNeT++ allows for the creation of a driver entity to build a network at run-time by program. One of the most important factors in any simulation is the programming language, which is C/C++ based. Since it possesses advantages and it is also freeware for the research community, we have selected the OMNeT++ simulator in our simulation studies

#### 4.3.2 FoF-R Ant Models

The internal structure of each FoF-R network node implemented in OMNeT++ is illustrated in Figure 4.7.

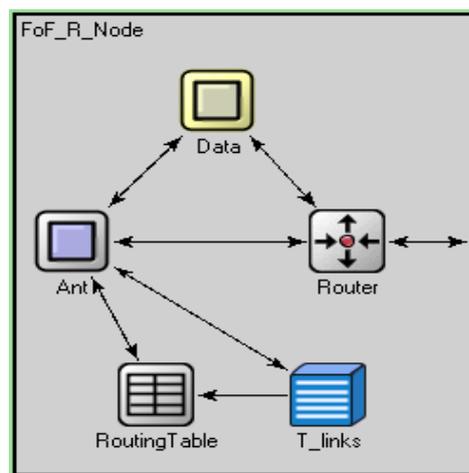


Figure 4.7 The FoF-R ant node structure

- **Router** is responsible for traffics queuing modeling and switching the FoF-R ant message and data traffic to other routers.
- **Data** is the sub-module that generates and receives the data traffic and collects related statistical values.
- **Ant** is the sub-module to handle the FoF-R ant agents e.g., ant generator and sink. It processes the FoF-R ant messages such as uploading and delivering link information and elapsed time into local node when an ant passing.
- **RoutingTable** holds the routing information of the node. In our study, the routing information has the next hop probability values.
- **$T\_link\_l$**  is the Distributed Resilience Matrix (DRM) database for recording the traversed links on the working and protection paths for each traffic flow, which is used to calculate next hop selection probability.

#### 4.3.3 FoF-R Ant Routing Parameter Setting

As other ant-based routing algorithms, the FoF-R ant-based routing algorithm has a collection of parameters to be set. We need to mention that, in any ant-based algorithms, the space of possible parameter settings is huge. The most common way for parameter setting is resolved through experiments. The values used in the simulations reported here were those found to be best according to our experiments.

The way of setting the pheromone parameters  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $a$  and  $b$  can affect the total capacity performance of FoF-R routing algorithm. Here,  $\beta$  is the parameter to adjust the weight of the path length on the amount of pheromone in equation (4.8),  $\gamma$  is the parameter to adjust the emphasize the capacity usage on the amount of pheromone in equation (4.9) and  $\alpha$  is a scalar parameter to adjust the importance of path length versus capacity usage in the amount of increased pheromone in equation (4.7).

The bigger value of  $\alpha$  is, the more emphasis on the path length and on the increased amount of pheromone  $\delta_r$ . On the other hand, the smaller value of  $\alpha$  is, the more emphasis on the capacity usage weight and the increased amount of pheromone  $\delta_r$ . For example, if  $\alpha = 1$ , a path with a shorter length is considered as a better solution regardless of its capacity usage. If  $\alpha = 0$ , the path with lower capacity usage is considered as a better routing solution regardless of its length. Moreover, for two paths that have their lengths that are nearly equal,

FoF-R solution will select the path with lower capacity usage. In our experiments, we reported that the value  $\alpha = 0.3-0.4$  is a good range for scalar parameter for FoF-R algorithm.

The selection of the values  $\beta$  and  $\gamma$  also depends on the setting of value  $\delta_r$ , which is the amount of adjusted pheromone. In fact,  $\delta_r$  will affect the performance and the stability of the ant-based algorithm. With a high value of  $\delta_r$ , the pheromone value of a route may change too fast and thus cause an instability on performance, In contrast, a small value of  $\delta_r$  make the pheromone value of a route change gradually, but we may need a large number of ants to update and increase the value of pheromone on a good route so that this route will become a good solution. Experiments show that the value of  $\delta_r$  should range between 0.05 and 0.2. The values of  $\beta$  and  $\gamma$  should be selected so that  $\delta_r$  ranges between 0.05 and 0.2. Because  $\delta_p$  and  $\delta_c$  are summed with the scale factor  $\alpha$  and  $1-\alpha$  in equation (4.7), we will select  $\beta$  and  $\gamma$  so that  $\delta_p$  and  $\delta_c$  take a value of around 0.2. In addition, the factors  $a$  and  $b$  used in Capacity Headroom-dependent Function  $R_l(h_l)$  tune the degree of attraction force to repulsion force, we found that  $a=10b$  can give a good range to be used in FoF-R algorithm.

However, determining an optimum set of parameters  $(\alpha, \beta, \gamma, a, b)$  for FoF-R ant based routing algorithm to achieve the best performance remains an open problem, which deserves further research effort.

In addition, the values for other general ant-based parameters are set as below:

- $\Delta t = 0.2$  second, it is time interval between two consecutive ant generations
- $TTL=2*N$ ,  $N$  is the number of nodes in the network, after which the ant is removed from the system
- $P_{noise} = 5\%$  or  $6\%$ , the exploration probability in the case of the 'local optima' condition

#### 4.4 Simulation Results

The FoF-R ant routing algorithm with DRM structure have been investigated by simulation on a PC with Intel(R) Celeron(R) 1.70GHz, 504MB of RAM, using the OMNeT++ discrete event simulator. All ILP models used herein were implemented in AMPL Mathematical Programming Language version 11.1 [98] and solved using the CPLEX 11.1 Solver [99].

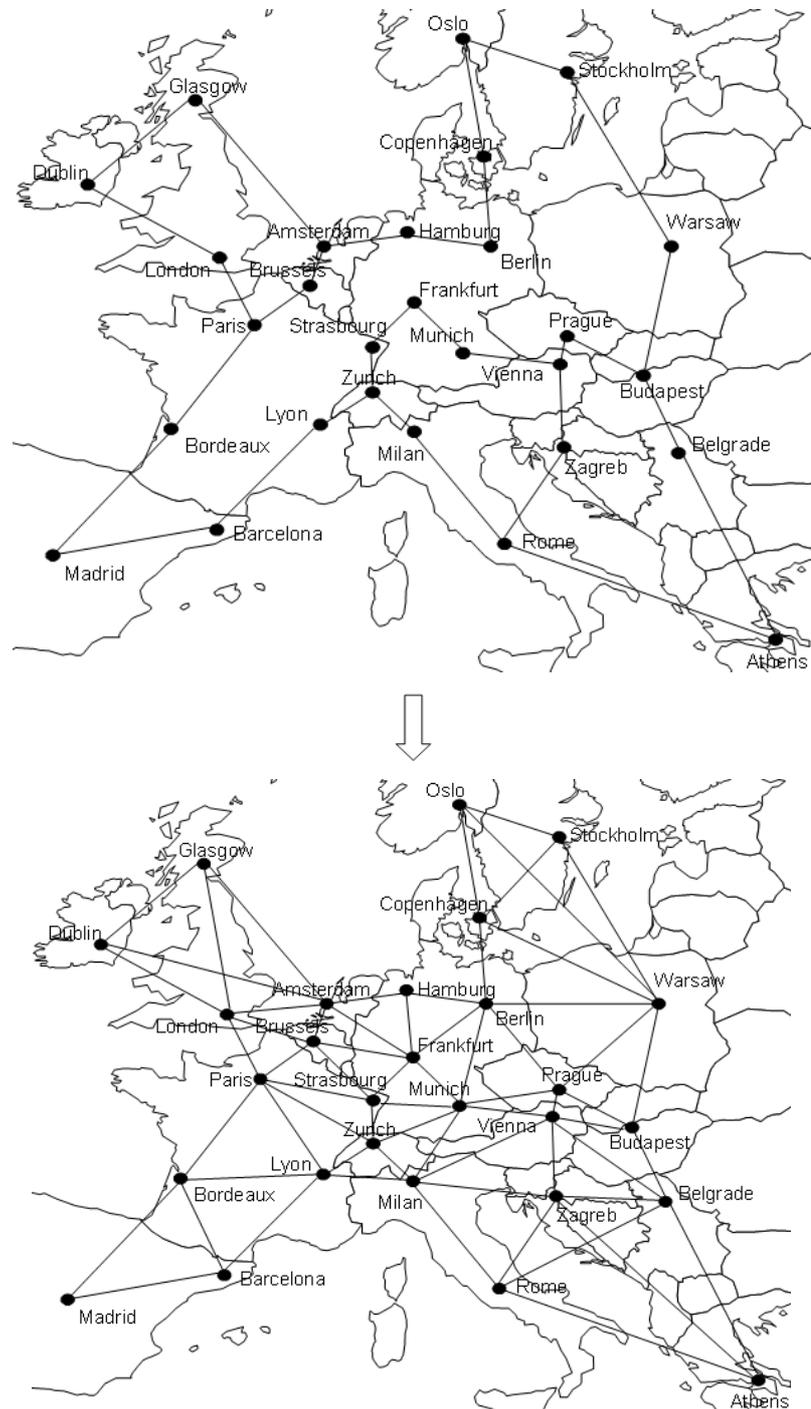


Figure 4.8 COST266 topology scenarios

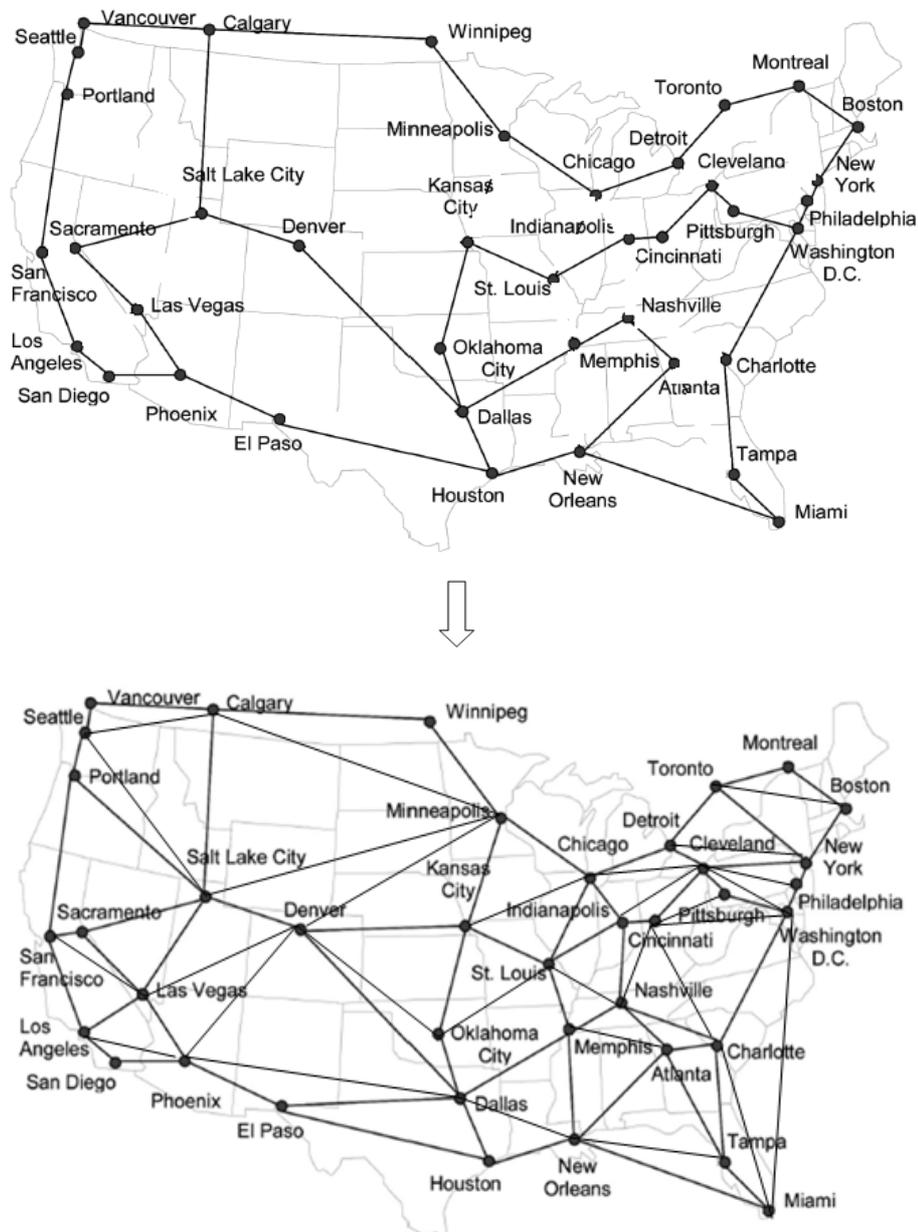


Figure 4.9 JANOS-US-CA network

The simulations are carried out upon two master network topologies from the SNDlib [100], one is the COST266 network with 28 nodes and 41 links and another is the JANOS-US-CA network with 39 nodes and 61 links. We have studied 12 versions of COST266 topologies and 16 versions of JANOS-US-CA network. Figure 4.8 depicts two limiting cases considered in COST266 networks: the sparsest topology with 31 links (upper case) and the densest topology with 64 links (lower case). Similarly, Figure 4.9 depicts two limiting cases

in JANOS-US-CA networks: the sparsest topology with 46 links (upper case) and the densest topology with 91 links (lower case). Without loss of generality, we assume symmetrical traffic flows, i.e., one unit of traffic demand between any pair of nodes. The comparative studies have been performed between our FoF-R algorithm with other three benchmarking mechanisms, i.e., SBPP, p-cycle and dedicated protection (DP), in terms of total capacity requirements and calculation time. The ILP models for SBPP and p-cycle were introduced in Chapter 2. For the DP scheme, by its very nature, it does not require optimization in the strictest sense, and each result was obtained by routing each demand via the shortest path and the next shortest disjoint path, so that the primary working routing will be equivalent to the SBPP design.

Table 4.1 Total capacity of four protection schemes in COST266 networks

<b>Links</b>	<b>Average node degree <math>\bar{d}</math></b>	<b>SBPP</b>	<b>FOF-R</b>	<b>P-Cycle</b>	<b>DP</b>
<b>31</b>	2.214	8264	8264	8372	13056
<b>34</b>	2.429	5584	5630	5987	8340
<b>37</b>	2.643	5362	5411	5717	7728
<b>40</b>	2.857	5029	5077	5316	7146
<b>43</b>	3.071	4787	4845	5063	6808
<b>46</b>	3.286	4641	4711	4814	6482
<b>49</b>	3.500	4482	4559	4676	6292
<b>52</b>	3.714	4184	4276	4349	5934
<b>55</b>	3.929	3980	4076	4076	5715
<b>58</b>	4.143	3886	4011	3885	5468
<b>61</b>	4.357	3696	3826	3655	5178
<b>64</b>	4.571	3481	3628	3448	5066

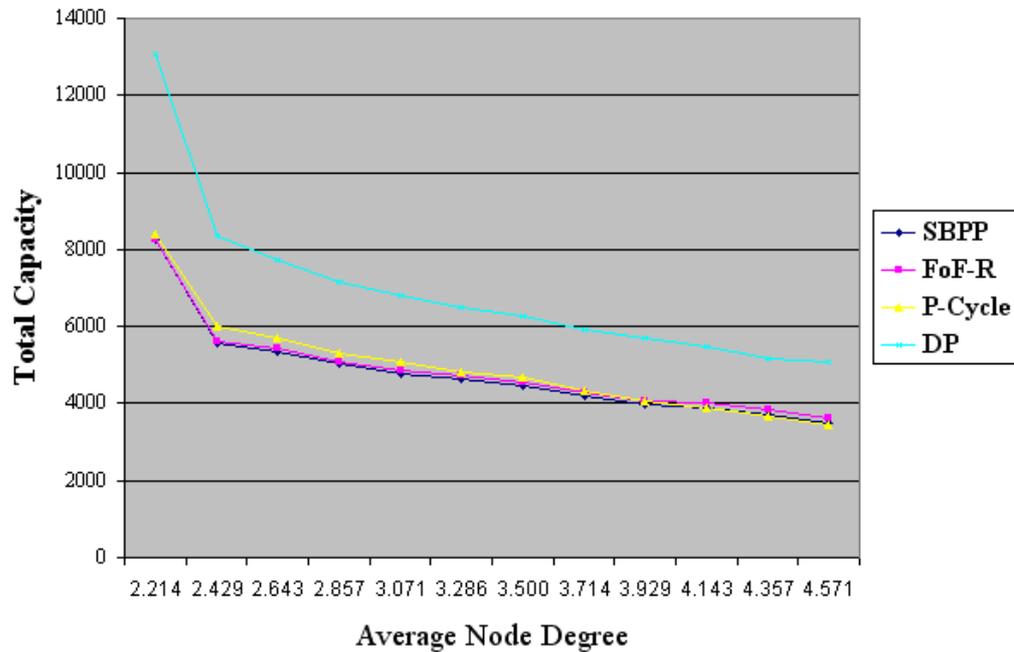


Figure 4.10 Total capacity usage comparison for COST266 networks

Table 4.1 and Figure 4.10 show the total capacity used by the four different protection schemes versus the average nodal degree  $\bar{d}$ . As the average nodal degree increases gradually from 2.214 to 4.571, the total capacity required for establishing restorable connections for all the requests decreases for all four protection schemes. The best total capacity allocation comes from the SBPP scheme in sparse networks i.e., when the average node degree is between 2.214 to 3.929, while p-cycle performs best in dense networks in which  $\bar{d}$  is greater than 4.143. The FoF-R scheme only uses 2% more capacity than SBPP in sparse networks and that of 5% more capacity than the p-cycle scheme in dense networks. DP provides no spare capacity sharing ability. Consequently, it gives the highest total capacity usage, generally exceeding the working capacity. It also suggests that the total capacity usage is dependent on the topological connectivity. All these four schemes tend to require less capacity when the network becomes denser. There are two possible reasons for the algorithm's sensitivity to network connectivity. First, as the network connectivity increases, both the predetermined working and protection path-pairs become shorter, this leads to a decrease in both working and protection capacity. Second, the potential for capacity sharing among PPs is likely to increase as the network connectivity increases, which leads to a decrease in protection capacity.

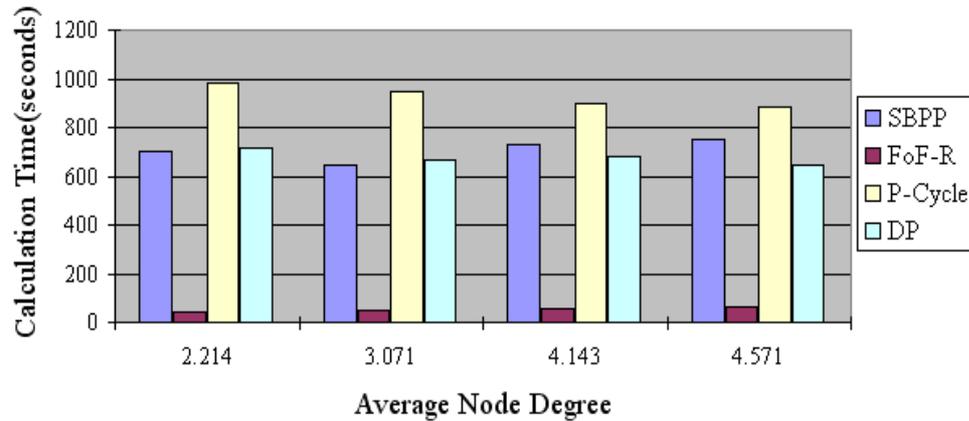


Figure 4.11 Algorithm calculation time for COST266 networks

Figure 4.11 shows that the calculation times for the different protection schemes. FoF-R is significantly different from the other three schemes. We can see that SBPP, p-cycle and DP find an optimal solution in hundreds of seconds with p-cycle always taking a longer time than SPBB, while FoF-R needs only in tens of seconds to find the near optimal solution. Thereby, it provides a good tradeoff between algorithm computation speed and capacity efficiency, especially in low-connectivity networks.

To validate our above findings, we conduct a similar comparative study based on a family of 16 topologies derived from the master JANOS-US-CA topology.

Table 4.2 Total capacity of four protection schemes in JANOS-US-CA networks

Links	Average node degree	SBPP	FoF-R	p-Cycle	DP
46	2.359	16738	17575	19250	26474
49	2.513	13760	14420	16130	21094
52	2.667	12552	13167	15040	19724
55	2.821	11635	12217	13818	18200
58	2.974	10973	11555	11783	15817
61	3.128	10161	10700	11188	15020
64	3.282	10169	10789	10960	14642
67	3.436	9807	10415	10069	13894
70	3.59	9581	10156	9714	13312
73	3.744	9310	9869	9448	13091
76	3.897	9012	9372	9114	12814
79	4.051	8515	9111	8561	12366
82	4.205	8018	8579	8007	11918
85	4.359	7850	8478	7839	11737
88	4.513	7909	8581	7730	11504
91	4.667	7837	8542	7533	11218

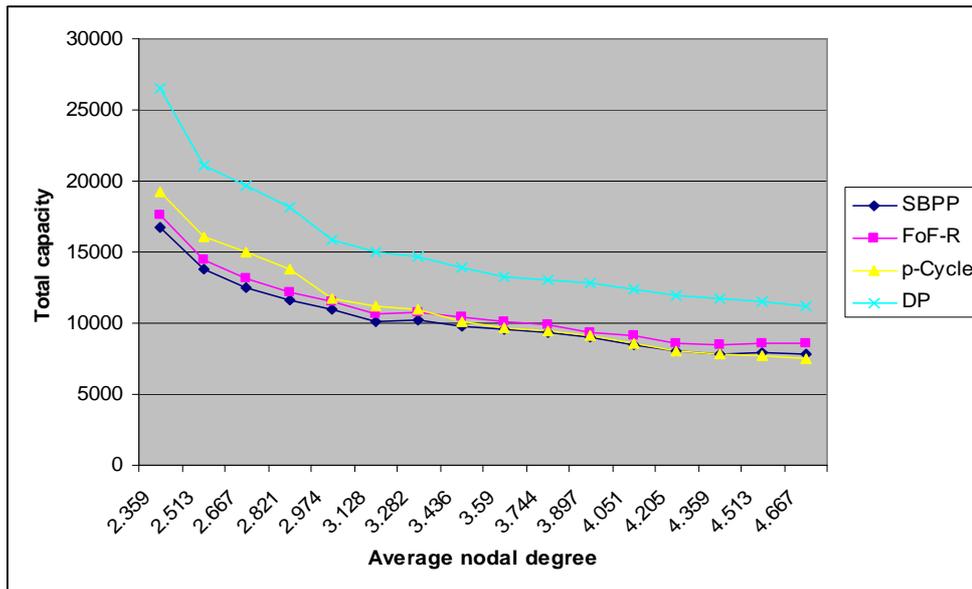


Figure 4.12 Total capacity usage comparison for JANOS-US-CA networks

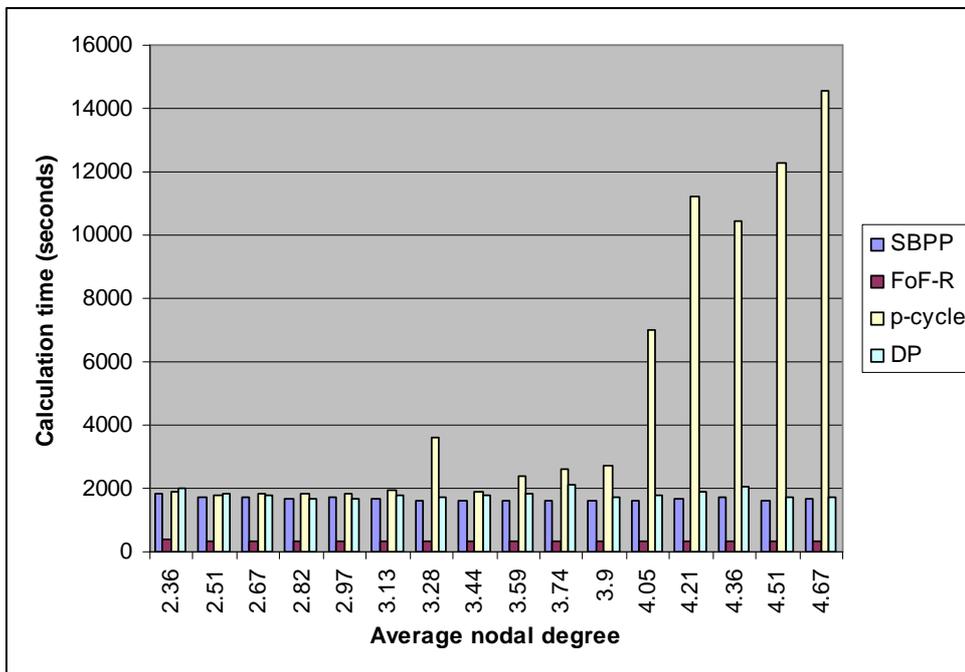


Figure 4.13 Algorithm calculation time for JANOS-US-CA network

Table 4.2 and Figure 4.12 show that the total capacity used by the four different protection schemes versus the average nodal degree  $\bar{d}$  on the larger JANOS-US-CA topologies. As the average nodal degree  $\bar{d}$  increases gradually from 2.359 to 4.667, the total capacity required for establishing restorable connections for all the requests decreases for all four protection schemes. The best total capacity allocation still comes from the SBPP scheme in sparse networks i.e., when the average node degree is  $\bar{d}$  between 2.359 to 4.051, while p-cycle performs best in dense networks in which  $\bar{d}$  is greater than 4.205. The FoF-R scheme is stable and continually approximates well the optimal solutions by SBPP, under a gap of 5% to 9% at most. Therefore, it can be seen that, the FoF-R ant based algorithm can approximate the optimal solution well. In terms of the calculation time, as seen in Figure 4.13, the FoF-R is significantly faster i.e., in the scale of tens of seconds, compared to the other three protections and the p-cycle scheme has the most sensitive relation between its calculation time and network connectivity.

Additionally, we have performed another comparative study only between SBPP and the FoF-R algorithm based on 10 different topologies from SNDlib, as shown in Figure 4.14 below.

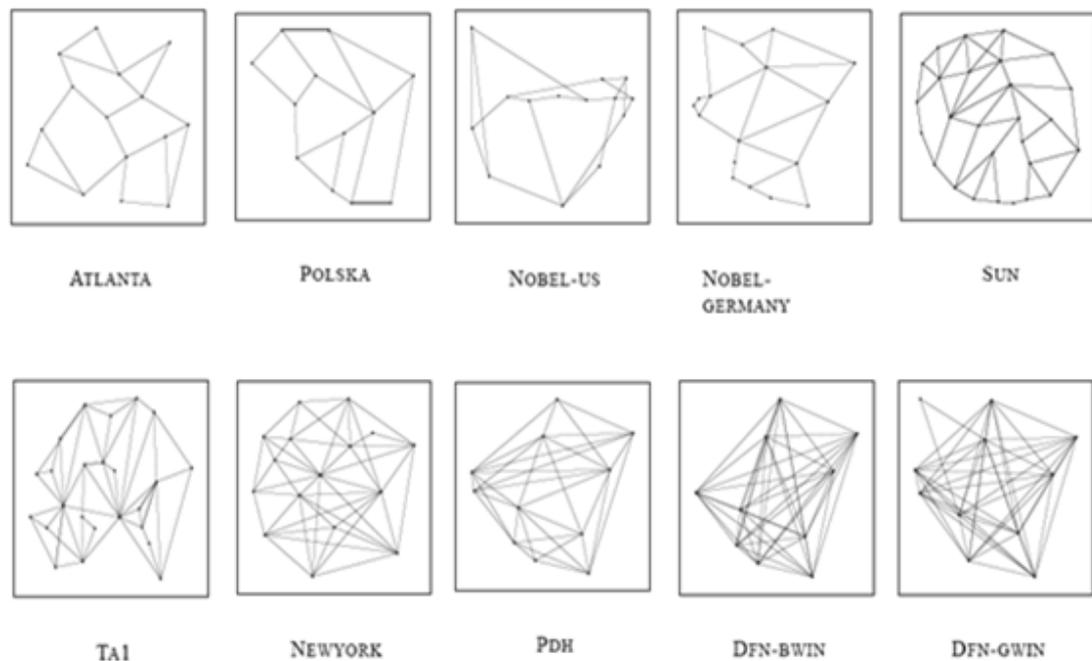


Figure 4.14 10 reference networks

Table 4.3 Total capacity usage comparison for 10 topologies

Network	$N$	$L$	Average node degree	Total capacity	
				SBPP	FoF-R
ATLANTA	15	22	2.93	997	1027
POLSKA	12	18	3.00	431	440
NOBEL-US	14	21	3.00	590	605
NOBEL-GERMANY	17	26	3.06	1437	1494
SUN	27	51	3.78	3155	3313
TAI	24	55	4.58	2265	2344
NEWYORK	16	49	6.13	578	595
PDH	11	34	6.18	213	230
DFN-GWIN	11	47	8.55	212	224
DFN-BWIN	10	45	9.00	148	148

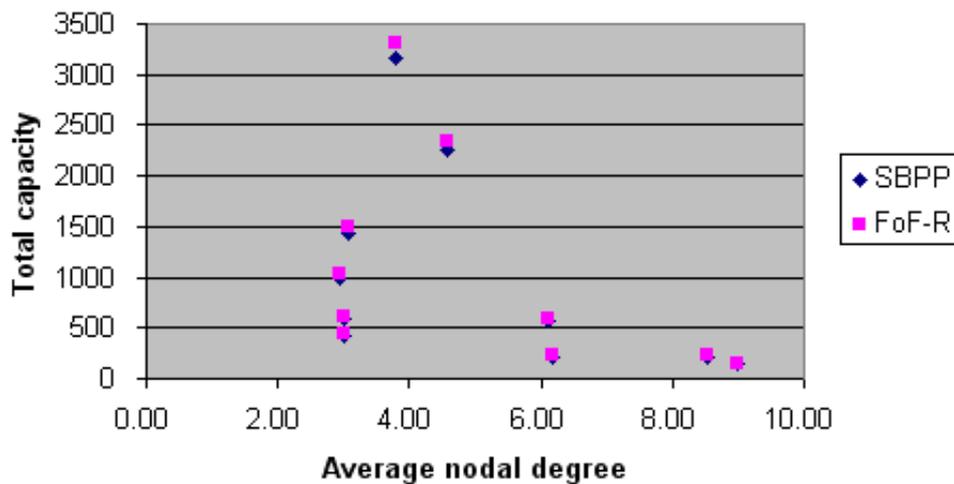


Figure 4.15 Total capacity usage vs. average nodal degree for 10 topologies

As shown in Table 4.3 and Figure 4.15, in terms of performance on total capacity, we found that the FoF-R ant based algorithm approximates well to the optimal solutions found by SBPP. While, we found that the average nodal degree metric does not work well to quantify the network connectivity, since there is no apparent correlation between the total capacity and average nodal degree. For example, there are around four significantly different solutions with average nodal degree about 3, i.e., 431 units in POLSKA network, 590 unit in NOBEL-US network, 1437 unit in NOBEL-GERMANY network and even 997 units in the ATLANTA network with the smallest average node degree of 2.93. The reason for that is

that the average nodal degree metric can only give a rough indication of network connectivity, without considering the shape and size of the network. Therefore, it is not critical to use this metric as a topological metric to measure network robustness. This finding becomes a strong driving force for us to explore a more accurate topological metric to quantify the network robustness and thus results in some novel findings presented in Chapter 5.

## **4.5 Summary**

In this chapter, we proposed a novel FoF-R ant-based routing algorithm to find the protection cycles and to explore the sharing potential among protection paths by using a Capacity Headroom Function (CHF), i.e., an attraction/repulsion relationship function. By using proactive ant mobile agents to continuously investigate the network capacity usage, and updating the DRMs at each node under a distributed control environment, our FoF-R algorithm showed good performance in trading off computational speed against capacity efficiency.



# Chapter 5

## Utility of Algebraic

## Connectivity Metric in SCA

Performance of survivable routing protocols, robustness of the network under failures and spare capacity allocation (SCA) depend crucially on the topology of the network. Network robustness can be characterized by the network topological connectivity, which expresses how well nodes are connected in a network. Most of publications in this area use the average nodal degree to reflect the effect of network connectivity in determining spare capacity allocation. Despite the wide adoption of the average nodal degree metric in such studies, we show that this metric is only a coarse indicator of how sparsely or densely connected a given topology is, and thus carries insufficient information on network topological structure. Furthermore, employing the average nodal degree for describing the network's connectivity may lead to misleading findings. We introduce a more informative metric: algebraic connectivity, which is defined as the 2<sup>nd</sup> smallest eigenvalue of the Laplacian matrix of a given topology in SCA. It is a more sensitive measure of network connectivity in a broader spectrum of graphs. It has desirable properties, such as the larger the algebraic connectivity is, the greater the number of node- and link-disjoint paths to choose from.

The remainder of this chapter is structured as follows. First, we review the methods to quantify the network robustness and then introduce the definition of algebraic connectivity metric and its related properties. Extensive simulation studies are presented next to compare

the algebraic connectivity metric with average nodal degree metric. Finally, conclusions are drawn.

## 5.1 Topological Measures of Network Robustness

Knowledge of network topology is crucial for understanding and predicting the performance, robustness, and scalability of network protocols. Routing and searching in networks, robustness to random network failures and spare capacity allocation (SCA) strategies all depend on the topological characteristics of networks. The robustness of a network can usually be characterized by a topological measure. In this section, we shall elaborate on the basic topological measures such as graph metrics. A variety of measures in both the structural domain and the spectral domain have been proposed to capture different features of a network topology as well as to classify graphs. We refer to [101] for a quite extensive survey of graph metrics. The structural measures refer to those measures such as node degree and clustering coefficient that represent topological properties more directly compared to spectral measures, which always involve an eigenvalue computation.

### 5.1.1 Structural Measures

In general, topological measures are a function of the network topology  $G(N,L)$ . The number of nodes  $N$  and the number of links  $L$  are mostly regarded as parameters of a network, not metrics. Many measures are highly correlated with the size of the network  $N$  and the number of Links  $L$ . The degree  $d_i$  of a node  $i$  in a network is the number of links that are incident upon the node  $i$ . The node degree is an important characteristic of a node e.g., it reflects the traffic capacity and the popularity of the node. In addition, the node degree distribution of a network, denoted as  $Pr[D = k]$ , expresses the fraction of nodes in the network with node degree  $k$ . In other words, it is the probability that a randomly chosen node has degree of  $k$ . The average nodal degree, denoted as  $\bar{d}$ , is purely an average function of the number of nodes  $N$  and the number of links  $L$ .

$$\bar{d} = E[D] = \sum_{k=1}^{d_{\max}} k \cdot Pr[D = k] = \frac{2L}{N} \quad (5.1)$$

where  $d_{\max}$  is the maximum node degree in the network.

The link density, denoted as  $p$ , is equal to the number of links  $L$  in the network divided by the maximal possible number of links that may exist in a network:

$$p = \frac{2L}{N(N-1)} = \frac{E(D)}{N-1} \quad (5.2)$$

The hopcount  $H_N$  of a shortest path is the number of links contained in that path. The hopcount distribution  $Pr[H_N = k]$  is the histogram of the hopcount between all possible node pairs in the graph. The average  $E[H_N]$  and the variance  $Var[H_N]$ , sometimes can be used to characterize the hopcount distribution. The largest hopcount between any pair of nodes is also referred to as the diameter of a network. In addition, the node connectivity  $\nu(G)$  and the link connectivity  $e(G)$  are the minimal number of nodes and links that have to be removed in order to disconnect a network. They seem natural quantifiers for robustness, but difficult to compute for large networks.

Among these structural metrics mentioned above, the average nodal degree  $\bar{d}$  is widely adopted to reflect the effect of the network connectivity on determining the amount of capacity allocation by most previous works [102-105]. Their simulation results have always shown how the total amount of working and spare capacity allocated to different network topologies varies according to their different average nodal degrees.

### 5.1.2 Spectral Measures

One of the main goals in graph theory is to deduce the principal properties and structure of a graph from its spectrum. The graph spectral analysis can be used to reveal the fundamental properties of a graph through geometric analytic and algebraic techniques. It has been shown that eigenvalues are closely related to almost all major invariants of a graph, linking one property to another [106].

Let  $G(N, L)$  be a network and  $N = |\mathcal{N}|$  is the number of nodes, where  $\mathcal{N}$  is the set of nodes; and  $L = |\mathcal{L}|$ , is the number of links, where  $\mathcal{L}$  denotes the set of links in the network. The network  $G$  can be represented by its adjacency matrix,  $A(G)$ , which is the  $N \times N$  matrix, whose  $(i, j)$ -th entry is “1” if node  $i$  is connected to node  $j$ , i.e.,  $(i, j) \in \mathcal{L}$ , and 0 otherwise. The diagonal entries of  $A(G)$ , are defined to be 0. Let  $D(G)$  be the  $N \times N$  diagonal matrix

with entries  $D_{i,i} = d_i$ , where  $d_i$  is the degree of the  $i$ -th node of  $G$ . The Laplacian matrix,  $Q(G)$ , of the network  $G$  is defined as:

$$Q(G) = D(G) - A(G) \quad (5.3)$$

The set of eigenvalues of the Laplacian matrix  $Q(G)$  for a given graph is called as the Laplacian spectrum of  $G$ . The second smallest eigenvalue  $\lambda_2$  is known as the algebraic connectivity or Fiedler value in [107-108], denoted as  $\lambda_2 = a(G)$  for simplicity. In previous works, the algebraic connectivity can be used to characterize network robustness regarding the following two dynamic processes: synchronization of dynamic processes at the nodes of a network and random walks on graphs. A network has a more robust synchronized state if the algebraic connectivity of the network is bigger. Random walks move and disseminate efficiently in topologies with large algebraic connectivity. In addition, the algebraic connectivity is also widely studied in various areas of mathematics, mainly in discrete mathematics and combinatorial optimization. In the following, we shall present some related mathematical results, which can reveal the topological implications of the algebraic connectivity and indicate the network robustness characteristic.

## 5.2 The Algebraic Connectivity Metric $a(G)$

The algebraic connectivity  $a(G)$  can characterize the robustness with respect to the topological connectivity of a network. It has been shown that the algebraic connectivity is only equal to zero if  $G$  is disconnected. In addition, the multiplicity of zero as an eigenvalue of the Laplacian matrix  $Q(G)$  is equal to the number of disconnected components of  $G$ . The network properties such as connectivity and cutsets have been studied in graph theory and a commonly agreed metric to reflect these properties is the algebraic connectivity. The higher the algebraic connectivity is, the more difficult it can be to be broken up into separate components.

### 5.2.1 The Derivation of Algebraic Connectivity

For the Laplacian matrix  $Q(G)$ , an  $N$ -dimensional vector  $\vec{x}$  is its eigenvector if there is a scalar  $\lambda$ , such that  $Q\vec{x} = \lambda\vec{x}$ . We denote  $\lambda$  is an eigenvalue of  $Q(G)$  corresponding to the eigenvector  $\vec{x}$ . By its definition, the  $Q(G)$  is a real symmetric and positive semi-definite matrix, thus all of its  $N$  eigenvalues are real and non-negative. Notice that the all-ones vector

is an eigenvector of any Laplacian matrix  $\mathbf{Q}$  and its associated eigenvalue is 0. We shall focus on the second smallest eigenvalue,  $\lambda_2$  of the Laplacian matrix and its associated eigenvector  $\vec{x}$ . Fiedler [109] named this eigenvalue  $\lambda_2$  as the ‘‘algebraic connectivity of a graph’’, and therefore, the algebraic connectivity  $\lambda_2$  is also known as the Fiedler value and its associated eigenvector as the Fiedler vector.

The following properties of algebraic connectivity and its Fiedler vector play an important role in quantifying network robustness.

- The algebraic connectivity  $a(G)$  of a network is greater than zero if and only if the network is connected.
- A Fiedler vector  $\vec{x} = (x_1, \dots, x_N)$  satisfies:

$$\sum_{i=1}^N x_i = 0, \quad (5.4)$$

Since all-ones vector is an eigenvector of the Laplacian matrix  $\mathbf{Q}(G)$  and the eigenvectors of a symmetric matrix are orthogonal. The Laplacian matrix  $\mathbf{Q}(G)$  has  $N$  nonnegative real eigenvalues:

$$0 = \lambda_1 \leq \lambda_2 = a(G) \leq \dots \leq \lambda_N \quad (5.5)$$

It can be seen that, 0 is always an eigenvalue of  $\mathbf{Q}(G)$ , and that  $\mathbf{1} = (1, 1, \dots, 1)^T$  is the corresponding eigenvector. Since  $\mathbf{Q}(G)$  is a symmetric matrix, then the Rayleigh quotient of  $\vec{x}$  with respect to  $\mathbf{Q}(G)$  [107] is:

$$\frac{\vec{x}^T \mathbf{Q}(G) \vec{x}}{\vec{x}^T \vec{x}} \quad (5.6)$$

Thus, the algebraic connectivity  $\lambda_2$  of the network  $G$  satisfies:

$$\lambda_2 = \min_{\vec{x} \perp (1, 1, \dots, 1)} \frac{\vec{x}^T \mathbf{Q}(G) \vec{x}}{\vec{x}^T \vec{x}} \quad (5.7)$$

In equation (5.7), the minimum value of  $\lambda_2$  occurs only when  $\vec{x}$  is the Fiedler vector.

For any vector  $\vec{x} \in R^N$ , we have:

$$\vec{x}^T \mathbf{Q}(G) \vec{x} = \sum_{(i,j) \in \mathcal{L}} (x_i - x_j)^2 \quad (5.8)$$

We denote the standard norm of a vector  $\vec{x}$  in Euclidean space by  $\|\vec{x}\| = \sqrt{x^T x}$  and the algebraic connectivity metric  $\lambda_2$  can be calculated from the following lemma.

**Lemma 5-1** Let  $G=(N,L)$  be a given network. Then  $\lambda_2$ , the algebraic connectivity of  $G$ , is give by:

$$\lambda_2 = \min \frac{\sum_{(i,j) \in \mathcal{L}} \|\vec{x}_i - \vec{x}_j\|^2}{\sum_{i=1}^N \|\vec{x}_i\|^2} \quad (5.9)$$

Where the minimum is taken over the vectors  $\{\vec{x}_1, \dots, \vec{x}_N\} \subset R^N$  such that  $\sum_{i=1}^N \vec{x}_i = \vec{0}$ , and the  $\vec{0}$  denotes the all-zeros vector. The magnitude of this value  $\lambda_2$  reflects how well connected the overall network  $G$  is.

### 5.2.2 The Relations between $a(G)$ , $e(G)$ and $v(G)$

We recall two traditional concepts in network connectivity:

- link connectivity  $e(G)$ , which is defined as the minimal number of links whose removal would result in losing connectivity of the network  $G$ ;
- node connectivity  $v(G)$ , which is defined as the minimal number of nodes together with adjacent links whose removal would result in losing connectivity of network  $G$ .

The algebraic connectivity  $a(G)$  is upper bounded by these two metrics and is illustrated to be a better robustness measure metric in [108]. In an incomplete graph, they have the following relations:

$$a(G) \leq v(G) \leq e(G) \leq d_{\min}(G) \quad (5.10)$$

The algebraic connectivity  $a(G)$  is a more useful robustness measure with respect to the network connectivity than the node and link connectivity. Unlike the traditional connectivity, the algebraic connectivity is dependent on the number of nodes, as well as the way in which nodes are connected. The node connectivity  $v(G)$  is always no smaller than the link connectivity  $e(G)$ , since deleting one node incident on each link in a cutset succeeds in disconnecting the network. Of course, smaller node subsets may be possible. The minimum node degree in network, denoted as  $d_{\min}(G)$ , is an upper bound on both the link and node

connectivity, since deleting all its neighbours (i.e., the links to all its neighbours) disconnects the network into one big and one single-node component.

### 5.2.3 Graph Partitioning

In addition to the relations to the intuitive node and link connectivity, the algebraic connectivity  $\lambda_2$  of a network is also closely linked to the cutset from the aspect of graph partitioning. It has been proved that networks with smaller algebraic connectivity have a better ratio cut [109-110]. A corollary of an extension of their work in [111] has demonstrated that one can obtain a good ratio cut from any vector with small Rayleigh quotient that is perpendicular to the all-ones' vector. Thus, the algebraic connectivity  $\lambda_2$  tells how well we can cut a graph. A cut of a graph is a division of its nodes into two sets,  $S$  and  $\bar{S}$ . We usually want to find that cut that has few links as possible. We let  $\mathcal{L}(S, \bar{S})$  denote the set of links whose nodes lie on opposite sides of the cut. We then define the ratio of the cut to be:

$$\phi(S) = \frac{|E(S, \bar{S})|}{\min(|S|, |\bar{S}|)} \quad (5.11)$$

The best cut is the one of minimum ratio, and its quality is the isoperimetric number  $\phi(G)$  of a graph [107]:

$$\phi(G) = \min_S \phi(S) \quad (5.12)$$

The Cheeger's inequality shows that the isoperimetric number  $\phi(G)$  is intimately related to  $\lambda_2$  as:

$$\phi \geq \lambda_2 \geq \frac{\phi^2}{2\Delta} \quad (5.13)$$

where  $\Delta$  is an upper bound on the node degree in the network. By Cheeger's inequality,  $\lambda_2$  gives an indication to how to quantify the network connectivity. If  $\lambda_2$  is small, then it is possible to cut the network into two pieces without cutting too many links. If  $\lambda_2$  is large, then every cut of the network must cut many links.

### 5.3 Properties of Algebraic Connectivity

In this section, we will introduce some linear properties of the algebraic connectivity,  $a(G)$ , which is related to measure the network topological robustness.

**Lemma 5-2** If  $G_1, G_2$  are link-disjoint networks with the same set of nodes then:

$$a(G_1 + G_2) \leq a(G_1 \cup G_2) \quad (5.14)$$

**Proof.** We denote  $X$  as the set of all column vectors  $x$  such that  $x^T x = 1$ , since  $Q(G_1 \cup G_2) = Q(G_1) + Q(G_2)$ , thus:

$$\begin{aligned} a(G_1 \cup G_2) &= \min_{x \in X} (x^T Q(G_1)x + x^T Q(G_2)x) \\ &\geq \min_{x \in X} x^T Q(G_1)x + \min_{x \in X} x^T Q(G_2)x = a(G_1) + a(G_2) \end{aligned}$$

Corollary, the function  $a(G)$  is non-decreasing for networks with the same set of nodes, and we have:

$$a(G_1) \leq a(G_2), \text{ if } G_1 \subseteq G_2 \quad (5.15)$$

where  $G_1, G_2$  have the same set of nodes.

**Lemma 5-3** Let  $G$  be a given network, and  $G_1$  arises from  $G$  by removing  $k$  nodes from  $G$  and all adjacent links then:

$$a(G_1) \geq a(G) - k \quad (5.16)$$

**Proof.** Let  $G$  have  $N$  nodes and let  $G_1$  arise from  $G$  by removing one node, say, define a new network  $G'$  by completing in  $G$  all missing links from deleted node, then:

$$Q(G') = \begin{pmatrix} Q(G_1) + I, & -e^T \\ -e, & N-1 \end{pmatrix}$$

Let  $\vec{x}$  be an eigenvector corresponding to the eigenvalue, since  $Q(G') \begin{pmatrix} \vec{x} \\ 0 \end{pmatrix} = [a(G_1) + 1] \begin{pmatrix} \vec{x} \\ 0 \end{pmatrix}$ , then  $a(G_1) + 1$  is an eigenvalue of  $Q(G')$  different from

zero, i.e.,  $a(G') \leq a(G_1) + 1$ . By equation (5.15),  $a(G') \leq a(G_1)$ , which implies equation (5.16) for  $k=1$ . The general case follows by induction.

#### 5.4 The Mean Distance

In [107], it has been shown that the algebraic connectivity  $\lambda_2$  is also closely related to some other graph invariants. One of the most interesting connections is its relation to the mean distance,  $\bar{\rho}(G)$  of graphs. The mean distance is equal to the average of all distances between distinct nodes of the graph. In some sense, this metric can measure the size and shape of the graph. In [112], some bounds on the mean distance  $\bar{\rho}(G)$  are derived. Its lower bound is:

$$(N-1)\bar{\rho}(G) \geq \frac{2}{\lambda_2} + \frac{N-2}{2} \quad (5.17)$$

and its upper bound is:

$$\bar{\rho}(G) \leq \frac{N}{N-1} \left( \left\lceil \frac{\Delta + \lambda_2}{4\lambda_2} \ln(N-1) \right\rceil + \frac{1}{2} \right) \quad (5.18)$$

Here  $G$  is a given network with  $N$  nodes,  $\lambda_2 = a(G)$ , is the algebraic connectivity of the network,  $\Delta = \Delta(G)$  is the maximal node degree. This information provided by the network mean distance  $\bar{\rho}(G)$ , is very useful to be considered when we comparing different network topologies in our later simulation work.

According to the above properties of algebraic connectivity metric, we can use this metric to measure the importance of a node or a link, because the larger the algebraic connectivity of a network is, the more connected the network will be. Thus, the topological connectivity of the remaining network can be quantified by the algebraic connectivity of the network resulting from removing that particular node and all the links connected to that node from the original network. In this way, the node or link that causes a more severe reduction in the remaining algebraic network connectivity has higher importance and should need more protection. In addition, we can also compare the effects of algebraic connectivity metric vs. average nodal degree metric, in terms of the performance on the total capacity allocated for different topologies. Therefore, we can propose a principle that both working and spare

capacity allocations benefit mostly from adding some critical nodes and links to maximize the algebraic connectivity of a current network.

## 5.5 Experimental Results

We have performed the following four experiments to investigate the effects of the topological metrics, i.e., algebraic connectivity vs. average nodal degree on spare capacity allocation.

### 5.5.1 Successive Deletion of Nodes

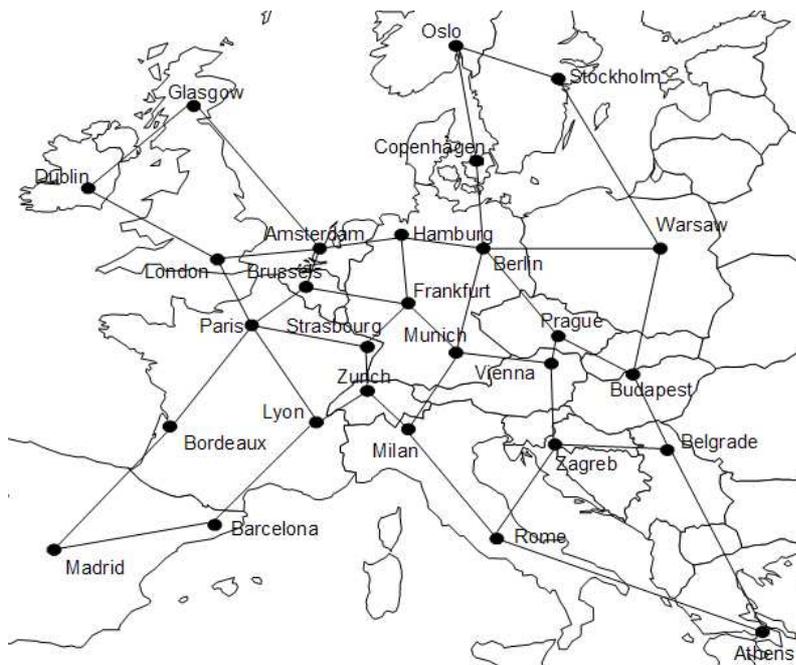


Figure 5.1 COST266 Network

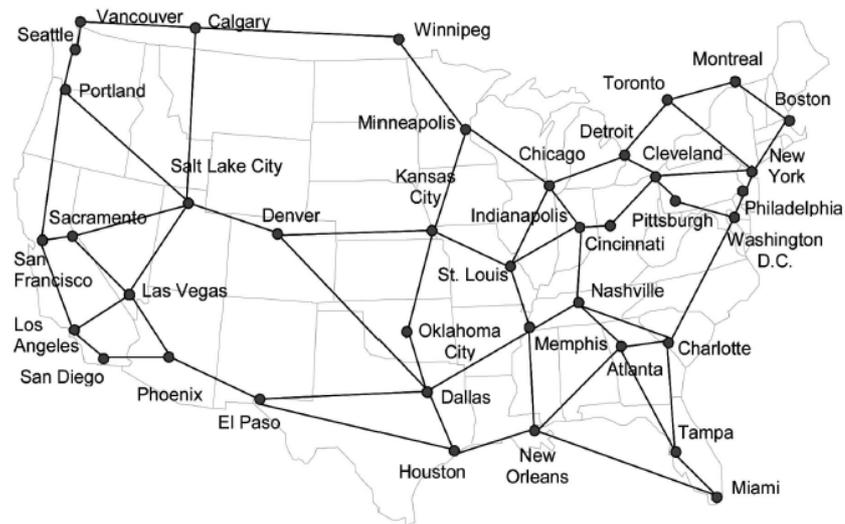


Figure 5.2 JANOS-US-CA Network

In the first experiment, two master network topologies from the SNDlib are considered. The first one is the COST266 network, see Figure 5.1. This network contains 28 nodes and 41 bidirectional links. The second network is JANOS-US-CA, as shown in Figure 5.2, which is based on the North-American Network with 39 nodes and 61 bidirectional links. The SBPP ILP model is solved using AMPL/CPLEX 11.1 on a PC with Intel(R) Celeron(R) 1.70GHz, 504MB of RAM.

Firstly, we investigate how the importance of each node affects the total capacity allocated. Simulations were conducted on two families of network topologies derived from the above two master networks by deleting one node each time, together with all of its adjacent links. Afterwards, we calculate the algebraic connectivity and average nodal degree of the remaining network. The SBPP algorithm is used to evaluate each topology alternative, to find the optimal total capacity. Here, we ignore some cases for which the SBPP cannot find feasible solutions since no node-disjoint paths exist for a traffic demand after the critical node has been deleted. For example, after deleting the “Oslo” node in Figure 5.1, its neighbouring nodes have only one adjacent link left, so there is no solution for the SBPP algorithm and it will be reported by the CPLEX solver. This is denoted as “impractical” in Total Capacity.

Table 5.1 Total capacity, algebraic connectivity and average nodal degree after deleting specific nodes in COST266 reference network

<b>Deleted Node</b>	<b>Average Nodal Degree</b>	<b>Algebraic Connectivity</b>	<b>Total Capacity</b>
Berlin	2.6667	0.0707	impractical
Paris	2.6667	0.0954	impractical
Hamburg	2.8148	0.1125	5694
Milan	2.8148	0.1128	5814
Zurich	2.8148	0.1197	impractical
Amsterdam	2.7407	0.1206	impractical
Frankfurt	2.7407	0.1297	5475
Munich	2.7407	0.1314	5616
Lyon	2.8148	0.1320	impractical
Rome	2.8148	0.1458	impractical
Warsaw	2.8148	0.1544	impractical
Copenhagen	2.8889	0.1632	impractical
Strasbourg	2.8148	0.1634	4685
Brussels	2.8148	0.1640	4724
Vienna	2.8148	0.1663	4853
Bordeaux	2.8889	0.1745	impractical
Prague	2.8148	0.1747	4572
Zagreb	2.8148	0.1761	4644
Barcelona	2.8889	0.1766	impractical
Athens	2.8889	0.1771	4671
London	2.8148	0.1773	impractical
Glasgow	2.8889	0.1774	impractical
Belgrade	2.8148	0.1787	impractical
Budapest	2.8148	0.1808	impractical
Dublin	2.8889	0.1822	impractical
Oslo	2.8889	0.1831	impractical
Stockholm	2.8889	0.1844	impractical
Madrid	2.8889	0.1891	impractical

Table 5.2 Total capacity, algebraic connectivity and average nodal degree after deleting specific nodes in JANOS-US-CA reference network

<b>Deleted Node</b>	<b>Average Nodal Degree</b>	<b>Algebraic Connectivity</b>	<b>Total Capacity</b>
Chicago	3.0000	0.0730	12160
Minneapolis	3.0526	0.0775	impractical
EI Paso	3.0526	0.0818	12403
Denver	3.0526	0.0846	11618
Salt Lake City	2.9474	0.0848	12396
Detroit	3.0526	0.0853	impractical
Charlotte	3.0000	0.0868	11254
Phoenix	3.0526	0.0877	impractical
Winnipeg	3.1053	0.0883	impractical
Calgary	3.0526	0.0888	impractical
New Orleans	3.0000	0.0910	impractical
Philadelphia	3.0526	0.0939	impractical
Dallas	2.9474	0.0956	impractical
Houston	3.0526	0.0959	10342
Kansas City	3.0000	0.0961	impractical
St. Louis	3.0000	0.0965	impractical
Memphis	3.0000	0.0978	10285
St Louis	3.0000	0.0979	10313
Cincinnati	3.1053	0.1034	10523
Cleveland	3.0000	0.1042	impractical
Nashville	3.0000	0.1056	10255
Toronto	3.0526	0.1058	impractical
Atlanta	3.0000	0.1068	9730
New York	3.0000	0.1069	impractical
Miami	3.1053	0.1090	9588
Tampa	3.0526	0.1094	impractical
Oklahoma	3.1053	0.1107	9638
Vancouver	3.1053	0.1120	impractical
Las Vegas	3.0000	0.1127	10068
Philadelphia	3.1053	0.1132	9724
San Diego	3.1053	0.1133	9594
Potland	3.0526	0.1142	impractical
Sacramento	3.0526	0.1145	9625
Pittsburgh	3.1053	0.1148	9659
Los Angeles	3.0526	0.1155	impractical
Seattle	3.1053	0.1163	impractical
San Francisco	3.0526	0.1165	9768
Montreal	3.1053	0.1169	impractical
Boston	3.1053	0.1173	impractical

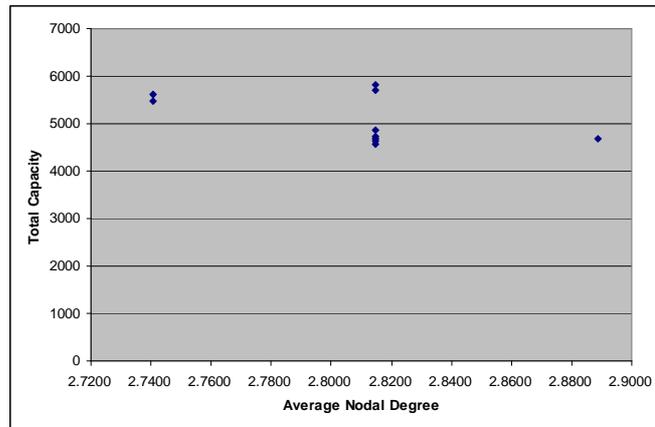


Figure 5.3 Total capacity vs. average nodal degree after deleting specific nodes in COST266 reference network

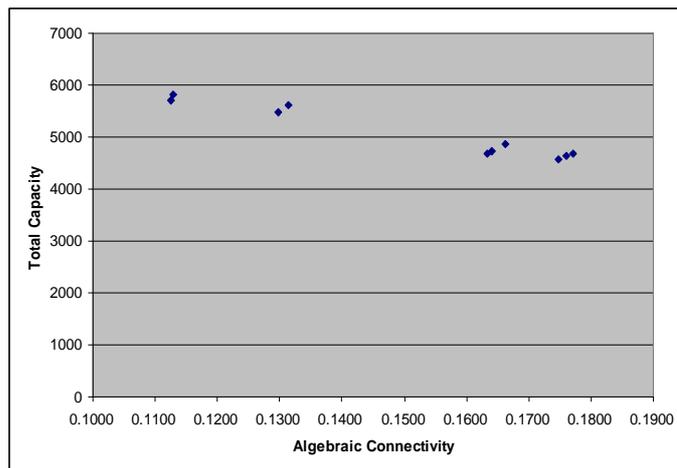


Figure 5.4 Total capacity vs algebraic connectivity after deleting specific nodes in COST266 reference network

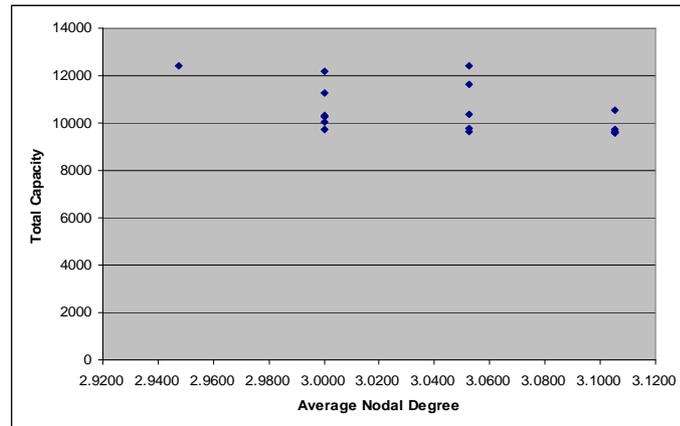


Figure 5.5 Total capacity vs. average nodal degree after deleting specific nodes in JANOS-US-CA reference network

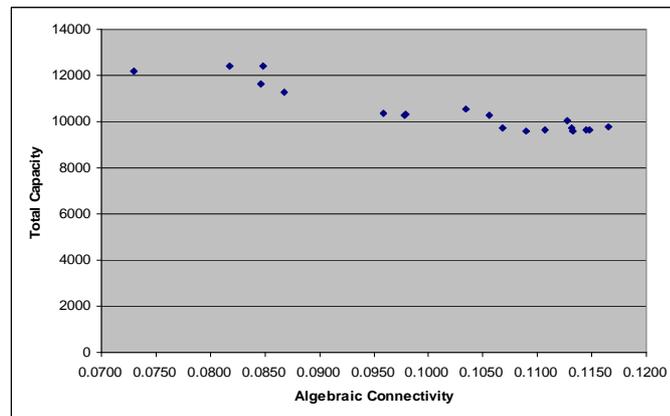


Figure 5.6 Total capacity vs. algebraic connectivity after deleting specific nodes in JANOS-US-CA reference network

It can be seen that, the results on the optimal total capacity, average nodal degree and algebraic connectivity of the remaining network after deleting specific nodes in the two reference networks are listed in Tables 5.1 and 5.2. These results have been depicted in Figures 5.3 to 5.6 for showing how the total capacity varies with the average nodal degree and the algebraic connectivity, respectively. It is evident that total capacity is more strongly correlated with the algebraic connectivity than with the average nodal degree. There are two possible reasons for the SBPP algorithm's sensitivity to network connectivity. Firstly, as the network connectivity increases, both the predetermined working and protection path-pairs become shorter. This leads to a decrease of both working and protection capacity. Secondly,

the potential for capacity sharing among protection paths is likely to increase as the network connectivity increases, and this leads to a further decrease of protection capacity

Looking at the results in detail, we can see almost linear dependence of the total capacity on the algebraic connectivity. By contrast, in the case of average nodal degree, its dependence on average nodal degree  $\bar{d}$  is not monotonic. For example, in Figure 5.3, there are 7 different topologies with  $\bar{d} = 2.8148$ , while they have 7 different total capacities allocated ranging from 4572 to 5814 units. This shows that using of average nodal degree as a metric has a severe limitation as it is insensitive to the total capacity of a given topology. On the other hand, algebraic connectivity monotonically depends on total capacity of a given topological structure. We also can see that, if the traffic demands are uniform, the nodes in the core region e.g., Hamburg, Milan, Frankfurt and Munich, are more important than others, because they are more frequently being used by traffic flows. If any of these nodes is deleted, it will result in a severe reduction of algebraic connectivity. The similar phenomenon can be observed in the results obtained for JANOS-US-CA network: see Figures 5.5 and 5.6.

### 5.5.2 Successive Deletion of Links

Further experiments have been carried out to analyze the properties of the algebraic connectivity metric and average nodal degree taking into account only slightly modified topological scenarios. We investigate how the importance of each link affects the total capacity allocated. Following the similar mechanism mentioned above, simulations were conducted on two families of network topologies derived from our two reference networks by deleting one link at a time. Here, we ignore cases for which the SBPP cannot find practical solutions since no node-disjoint paths existed for a given traffic demand after the critical links have been deleted, e.g., if the link between Oslo and Stockholm is deleted, see Figure 5.1. The simulation results are shown in Tables 5.3 to 5.4 and Figures 5.7 to 5.8.

Table 5.3 Total capacity, algebraic connectivity and average nodal degree after deleting specific links in COST266 reference network

Deleted Link		Algebraic Connectivity	Total Capacity
Hamburg	Berlin	0.1176	6110
Zurich	Milan	0.1198	6015
Copenhagen	Hamburg	0.1258	impractical
Bordeaux	Paris	0.1309	impractical
Amsterdam	Hamburg	0.1361	5402
Milan	Rome	0.1364	5778
Barcelona	Lyon	0.1401	impractical
Frankfurt	Munich	0.1402	5680
Lyon	Zurich	0.1449	impractical
Stockholm	Warsaw	0.1527	impractical
Munich	Vienna	0.1589	5545
Berlin	Warsaw	0.1589	4969
Brussels	Frankfurt	0.1591	4934
Glasgow	Amsterdam	0.1595	impractical
Oslo	Copenhagen	0.1611	impractical
Frankfurt	Strasbourg	0.1635	5030
Paris	Strasbourg	0.1639	4906
Berlin	Munich	0.1646	5029
Rome	Athens	0.1663	impractical
Paris	Brussels	0.1680	4922
Madrid	Bordeaux	0.1687	impractical
London	Amsterdam	0.1689	4911
Berlin	Prague	0.1707	4930
Madrid	Barcelona	0.1709	impractical
Prague	Budapest	0.1714	4909
Zagreb	Rome	0.1719	4965
Munich	Milan	0.1722	5091
Vienna	Zagreb	0.1722	4939
Hamburg	Frankfurt	0.1723	4855
Dublin	London	0.1724	impractical
Zagreb	Belgrade	0.1727	4828
Prague	Vienna	0.1729	5095
Strasbourg	Zurich	0.1733	4886
Dublin	Glasgow	0.1735	impractical
Belgrade	Athens	0.1741	impractical
Budapest	Belgrade	0.1747	5307
Amsterdam	Brussels	0.1748	5026
Paris	Lyon	0.1748	4918
London	Paris	0.1748	impractical
Warsaw	Budapest	0.1748	impractical
Oslo	Stockholm	0.1750	impractical

Table 5.4 Total capacity, algebraic connectivity and average nodal degree after deleting specific links in JANOS-US-CA reference network

Deleted Link		Algebraic connectivity	Total capacity
Phoenix	EI Paso	0.0821	13106
Winnieg	Minneapolis	0.0826	impractical
Chicago	Detroit	0.0828	12159
Salt Lake City	Denver	0.0847	12162
Washington D.C.	Charlotte	0.0872	11901
Calgary	Winnipeg	0.0880	impractical
Minneapolis	Chicago	0.0897	11604
Houston	New Orleans	0.0970	11062
Indianapolis	Cincinnati	0.0976	impractical
Dallas	Memphis	0.0995	10598
Detroit	Toronto	0.0995	impractical
Kansas City	St. Louis	0.1006	10612
Vancouver	Calgary	0.1016	impractical
Denver	Kansas City	0.1019	10410
Cincinnati	Cleveland	0.1032	impractical
EI Paso	Houston	0.1035	10525
Philadelphia	Washington D.C.	0.1049	impractical
Memphis	Nashville	0.1055	10305
San Diego	Phoenix	0.1058	impractical
Denver	Dallas	0.1059	10312
New Orleans	Atlanta	0.1065	10305
St. Louis	Indianapolis	0.1066	10628
Cleveland	New York	0.1067	10582
EI Paso	Dallas	0.1068	10558
Portland	Salt Lake City	0.1070	10470
Toronto	Montreal	0.1070	impractical
New Orleans	Miami	0.1075	impractical
Pittsburgh	Washington D.C.	0.1076	impractical
Las Vegas	Phoenix	0.1080	10331
Nashville	Charlotte	0.1087	10624
Sacramento	Salt Lake City	0.1090	10354
Chicago	St. Louis	0.1090	10199
Tampa	Miami	0.1090	impractical
Atlanta	Charlotte	0.1091	10214
Vancouver	Seattle	0.1092	impractical
Detroit	Cleveland	0.1095	10206
Charlotte	Tampa	0.1097	10348
Los Angeles	Las Vegas	0.1100	10206
Las Vegas	Salt Lake City	0.1100	10281
Los Angeles	San Diego	0.1101	impractical
New York	Philadelphia	0.1101	impractical
San Francisco	Sacramento	0.1102	10359
Chicago	Indianapolis	0.1102	10342
Seattle	Portland	0.1102	impractical
Calgary	Salt Lake City	0.1103	10621
Dallas	Houston	0.1104	10093
St. Louis	Memphis	0.1104	10291
Kansas City	St. Louis	0.1104	impractical
Sacramento	Las Vegas	0.1105	10409
Cleveland	Pittsburgh	0.1105	impractical
Cleveland	New York	0.1105	impractical
San Francisco	Los Angeles	0.1106	10455
Minneapolis	Kansas City	0.1106	10320
Memphis	New Orleans	0.1106	10172
Indianapolis	Nashville	0.1106	10232
Atlanta	Tampa	0.1106	10358
Oklahoma City	Dallas	0.1106	impractical
Portland	San Francisco	0.1107	10348
Nashville	Atlanta	0.1107	10377
Toronto	New York	0.1107	10199
Montreal	Boston	0.1107	impractical

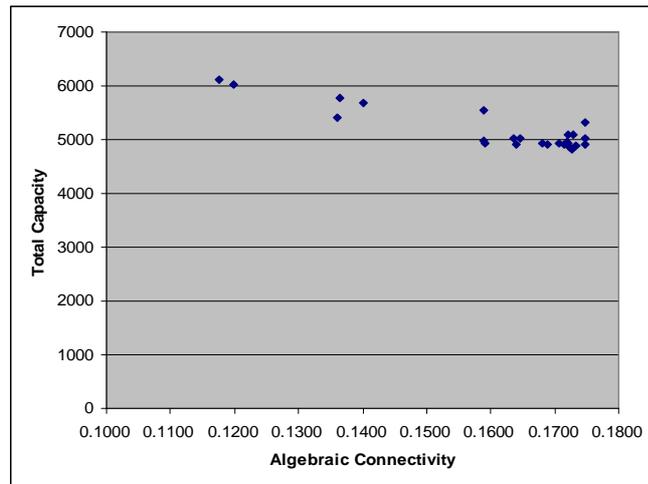


Figure 5.7 Total capacity vs algebraic connectivity after deleting specific links in COST266 reference network

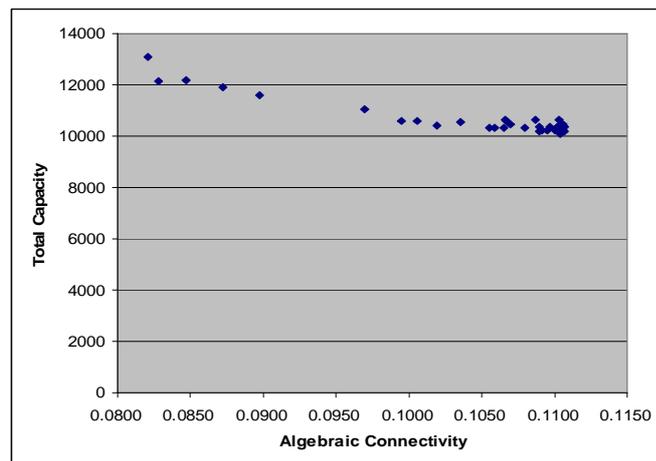


Figure 5.8 Total capacity vs algebraic connectivity after deleting specific links in JANOS-US-CA reference network

From the above results, it can be seen that the average nodal degree in two families of network topologies assumes constant value of  $\bar{d}=2.8571$  in COST 266 topologies with 40 links, and the value of  $\bar{d}=3.0769$  in JANOS-US-CA topologies with 60 links, respectively, while the total capacity solutions are significantly different. There are 25 solutions, with the total capacity ranging from 4828 to 6110 units in COST266 scenarios, and 40 solutions with the total capacity ranging from 10093 to 13106 units in JANOS-US-CA scenarios. This shows again that the average nodal degree has a severe limitation as it is insensitive to

changes in total capacity caused by removal of single link. The algebraic connectivity remains sensitive to such changes. In addition, it can be seen that the links located in the network's core region are more important than those at the network boundaries since they are more frequently used by the traffic flows. Thus, deleting them can cause severe decreases in network connectivity.

### 5.5.3 Repositioning of Links

Additionally, we investigated the impact of the algebraic connectivity metric and average nodal degree has on capacity allocation under links' repositions scenario. Seven sample networks derived from the COST266 reference network by placing 4 links in different positions have been studied, see Figures 5.9 to 5.15 below.

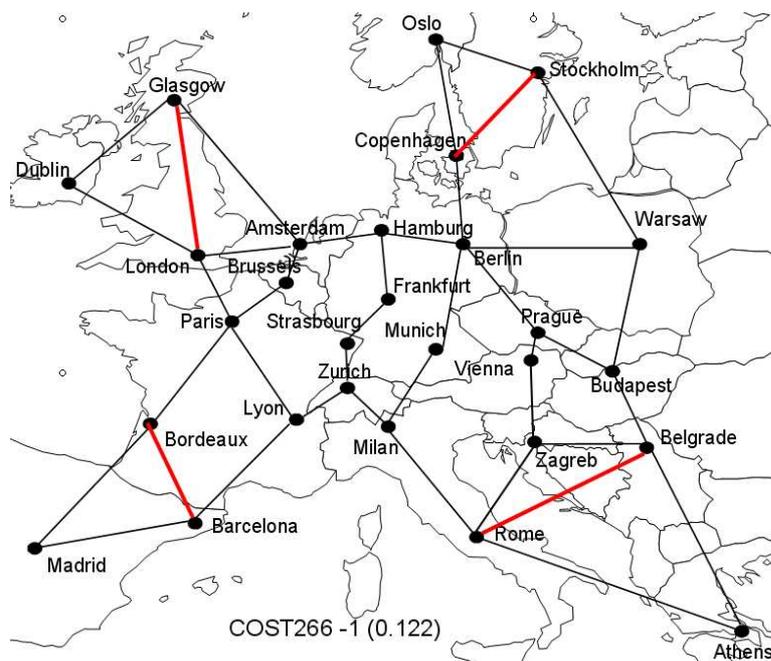


Figure 5.9 COST266-1

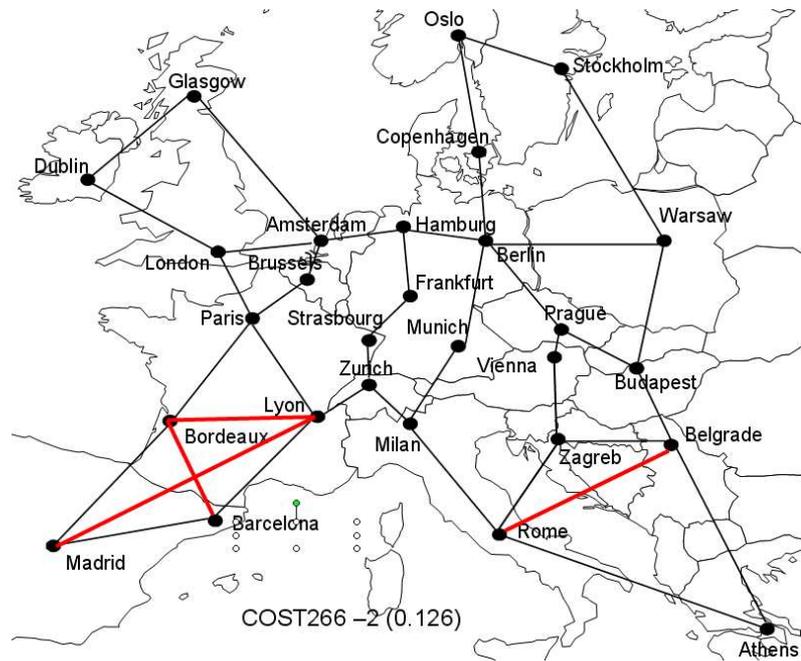


Figure 5.10 COST266-2

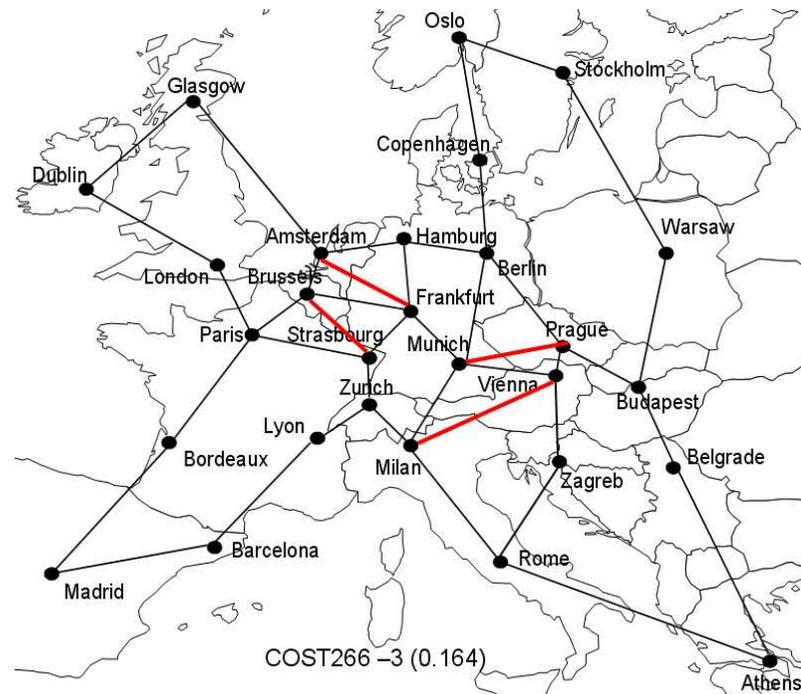


Figure 5.11 COST266-3

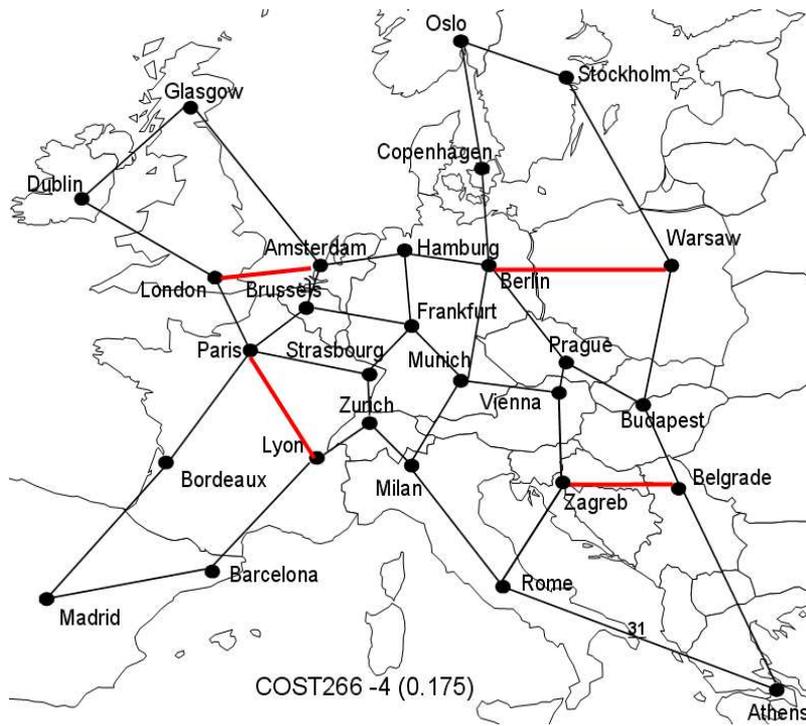


Figure 5.12 COST266-4

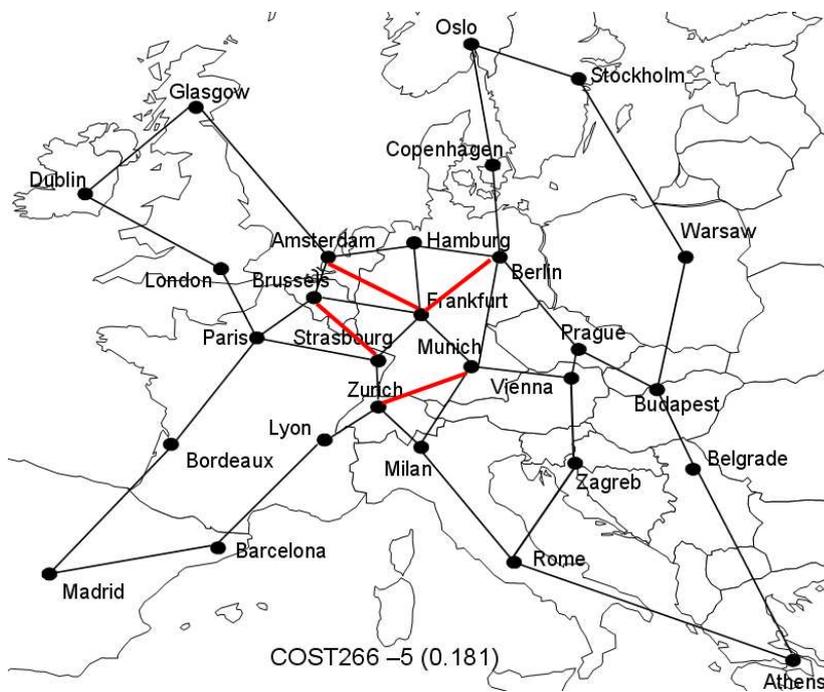


Figure 5.13 COST266-5

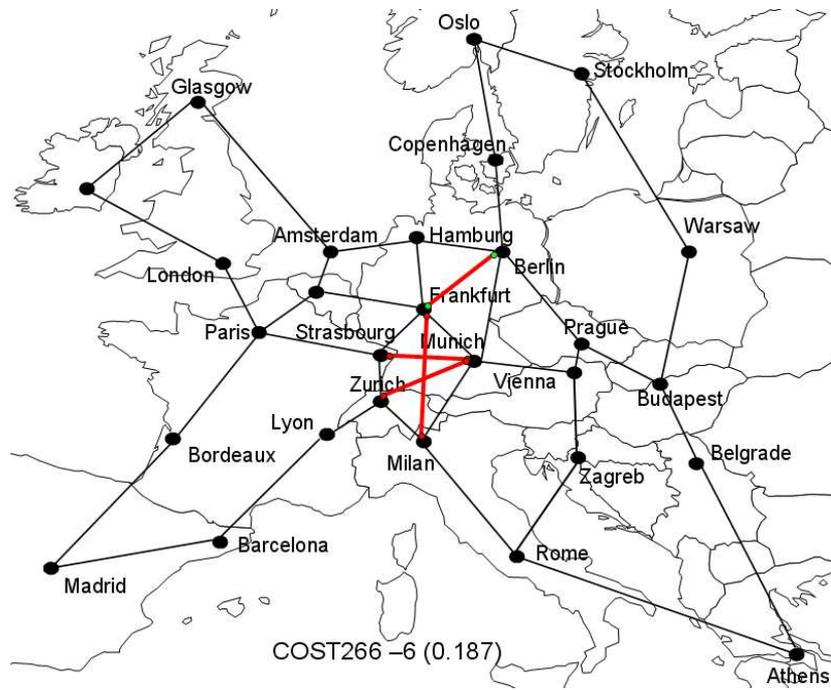


Figure 5.14 COST266-6

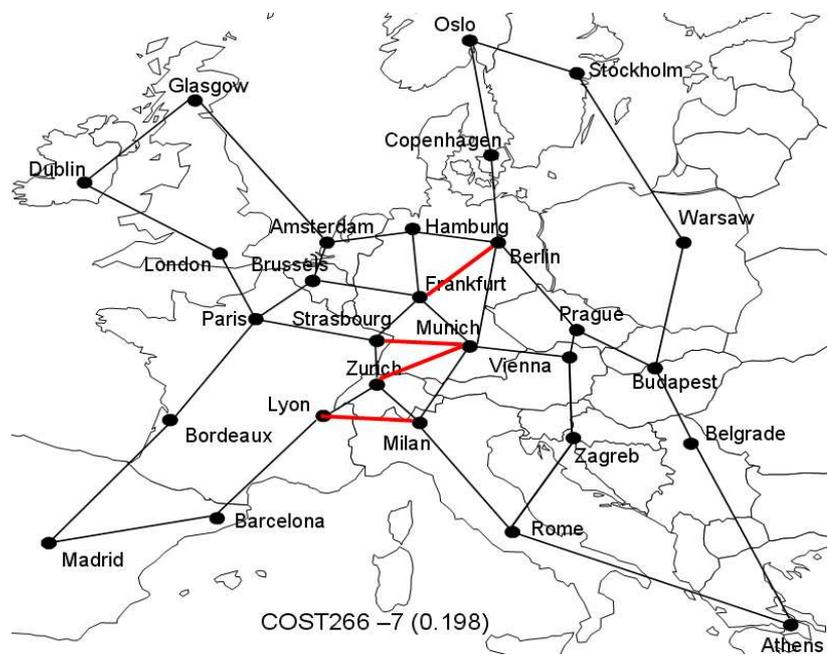


Figure 5.15 COST266-7

Table 5.5 Total capacity vs. algebraic connectivity for link repositions in COST266 reference network

Topology	Algebraic Connectivity	Total Capacity
COST266-1	0.122	5794
COST266-2	0.126	5820
COST266-3	0.164	4951
COST266-4	0.175	4778
COST266-5	0.181	4925
COST266-6	0.187	4751
COST266-7	0.198	4571

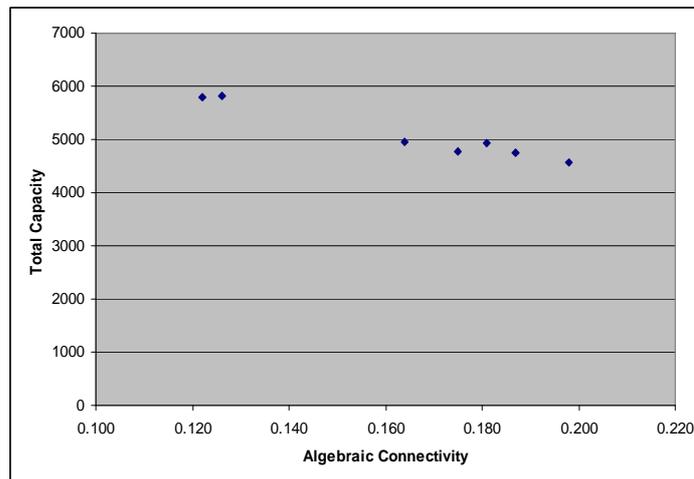


Figure 5.16 Total capacity vs algebraic connectivity for link repositions in COST266 reference network

As shown in Table 5.5 and Figure 5.16, while all the seven derived topologies have the same average nodal degree, i.e.,  $\bar{d} = 2.9286$ , the resulting total capacity values are quite different for each of them. Note that total capacity decreases as algebraic connectivity increases. One can see that when four links are placed on the boundary of the network, see e.g., Figure 5.10 and 5.11, the total capacity is generally larger than deploying the links in the core region of the network, cf. Figures 5.14 and 5.15, because boundary links are less used in the SBPP solutions.

### 5.5.4 The Combined Metric $\frac{\rho}{\lambda_2}$

In the previous three experiments, the comparative studies have been performed among the derived topologies from the same master topology and thus the topologies all look similar. In this case, we do not need to consider the size and shape of the topology and only the algebraic connectivity metric seems sufficient. In this experiment, by introducing the additional mean distance metric  $\bar{\rho}$ , which is also related to algebraic connectivity  $\lambda_2$ , we propose to compare different topologies, in terms of their size and shape. We also calculate the exact mean distance as a reference to the upper bound on the mean distance predicted in equation (5.17) and (5.18). We have selected 20 topology instances from the SNDlib, as shown in Figure 5.17 and their network information is listed in Table 6.

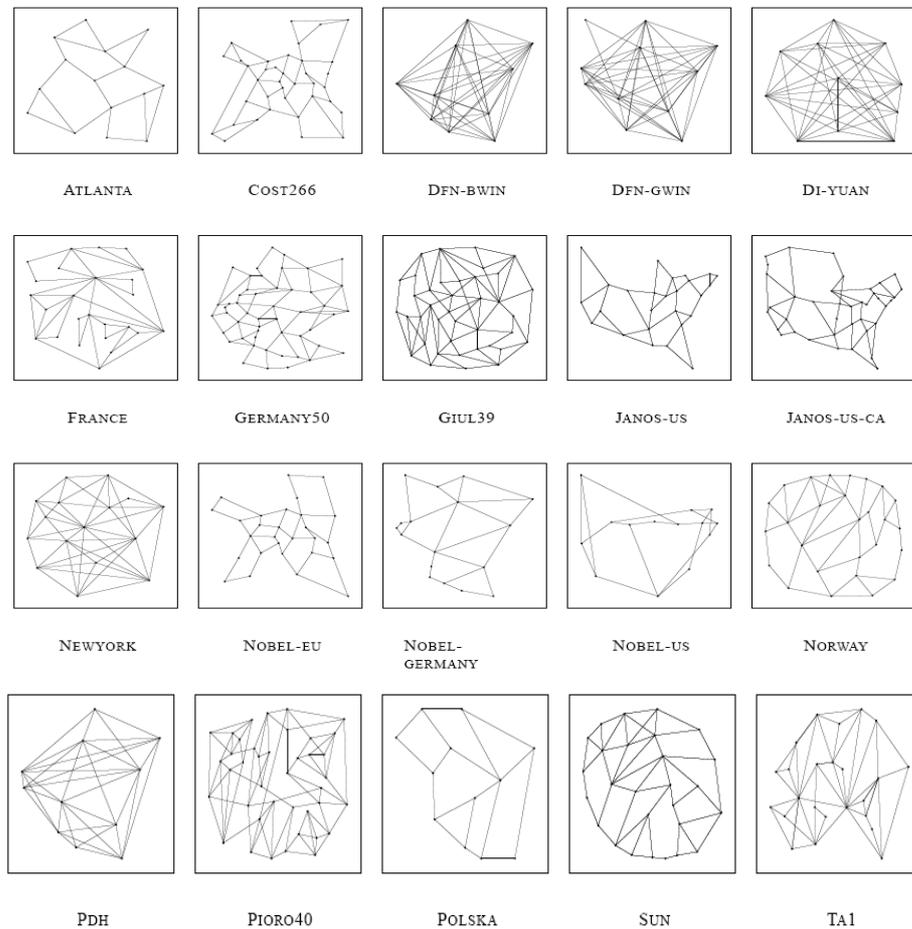


Figure 5.17 20 referenced network topologies

Table 5.6 Network Information for 20 referenced networks

<b>No.</b>	<b>Network</b>	<b><i>N</i></b>	<b><i>L</i></b>
1	ATLANTA	15	22
2	COST266	28	41
3	DFN-BWIN	10	45
4	DFN-GWIN	11	47
5	DI-YUAN	11	42
6	FRANCE	25	45
7	GERMANY50	50	88
8	GIUL39	39	86
9	JANOS-US	26	84
10	JANOS-US-CA	39	61
11	NEWYORK	16	49
12	NOBEL-EU	37	57
13	NOBEL-GERMANY	17	26
14	NOBEL-US	14	21
15	NORWAY	27	51
16	PDH	11	34
17	PIORO40	40	89
18	POLSKA	12	18
19	SUN	27	51
20	TA1	24	55

We also recall the lower and upper bounds on mean distance in equation (5.17-18), such that:

$$\left(\frac{2}{\lambda_2} + \frac{n-2}{2}\right)/(n-1) \leq \rho \leq \frac{n}{n-1} \left( \left\lceil \frac{\Delta + \lambda_2}{4\lambda_2} \ln(n-1) \right\rceil + \frac{1}{2} \right)$$

Then we calculate the combined metric  $\frac{\rho}{\lambda_2}$  to study its effects on the total capacity.

Here we use two mean distance related parameters, one is the exact mean distance  $\rho_{exact}$ , which can be calculated by the standard graph toolbox in MATLAB and another is the estimated upper bound on the mean distance  $\rho_{up}$ .

Table 5.7 Total capacity vs. algebraic connectivity, exact mean distance and upper bound of mean distance for 20 reference networks

No.	$N$	$L$	Average Node Degree	$\lambda_2$	$\Delta$	Total Distance	$\rho_{\text{exact}}$	$\rho_{\text{exact}}/\lambda_2$	$\rho_{\text{up}}$	$\rho_{\text{up}}/\lambda_2$	Working capacity	Spare capacity	Total capacity
1	15	22	2.933	0.426	4	526	2.505	5.887	7.352	17.279	526	471	997
2	28	41	2.929	0.175	5	2692	3.561	20.348	25.268	144.389	2774	2085	4859
3	10	45	9.000	10.000	9	90	1.000	0.100	1.160	0.116	90	58	148
4	11	47	8.545	2.000	10	126	1.145	0.573	3.799	1.900	126	86	212
5	11	42	7.636	5.794	9	136	1.236	0.213	1.617	0.279	136	57	193
6	25	45	3.600	0.354	10	1556	2.593	7.334	24.233	68.533	1566	1305	2871
7	50	88	3.520	0.183	5	9918	4.048	22.145	28.148	153.985	9924	3700	13624
8	39	86	4.410	0.377	8	4540	3.063	8.117	20.718	54.896	4540	1494	6034
9	26	42	3.231	0.197	5	2150	3.308	16.824	22.121	112.520	2162	1582	3744
10	39	61	3.128	0.111	5	6232	4.205	37.987	43.089	389.242	6282	4130	10412
11	16	49	6.125	1.503	11	412	1.717	1.142	6.008	3.998	418	160	578
12	37	57	3.081	0.162	5	4964	3.727	23.033	29.375	181.549	5038	3640	8678
13	17	26	3.059	0.302	6	734	2.699	8.941	15.378	50.954	750	687	1437
14	14	21	3.000	0.707	5	388	2.132	3.015	5.574	7.882	390	200	590
15	27	59	4.370	0.344	7	2006	2.858	8.307	18.058	52.494	2006	804	2810
16	11	34	6.182	2.639	8	157	1.427	0.541	2.553	0.967	154	59	213
17	40	89	4.450	0.219	5	5170	3.314	15.154	22.416	102.495	5170	2366	7536
18	12	18	3.000	0.713	5	282	2.136	2.998	5.243	7.359	282	149	431
19	27	51	3.778	0.286	6	2198	3.131	10.959	18.610	65.137	2200	955	3155
20	24	55	4.583	0.472	11	1278	2.315	4.905	19.880	42.120	1278	987	2265

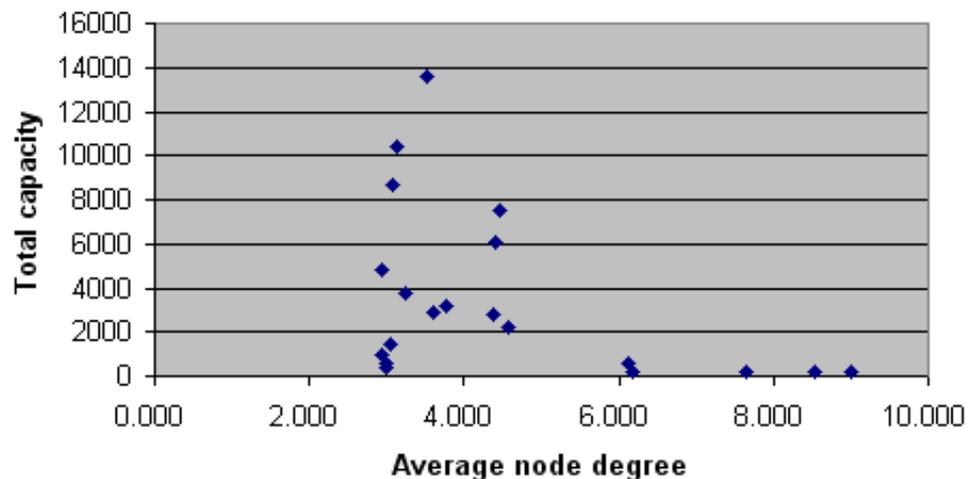


Figure 5.18 Total capacity vs. average nodal degree

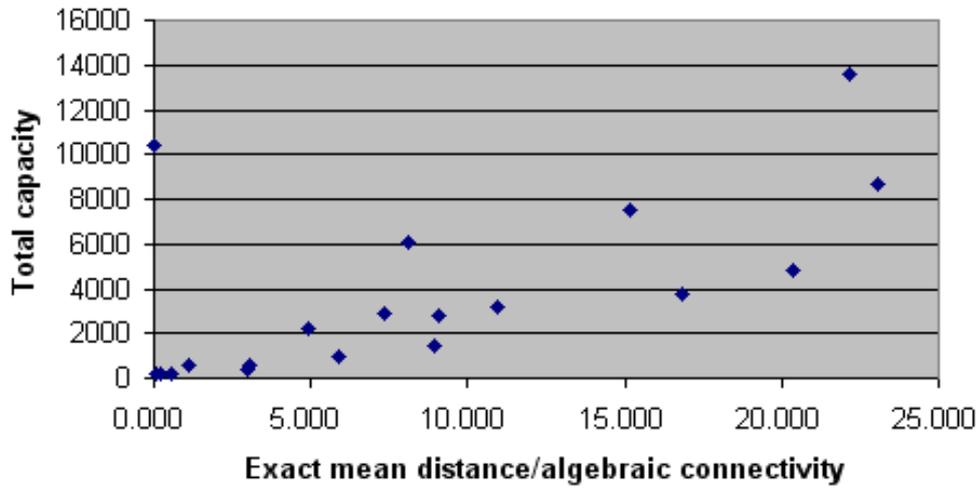


Figure 5.19 Total capacity vs. Exact mean distance/algebraic connectivity

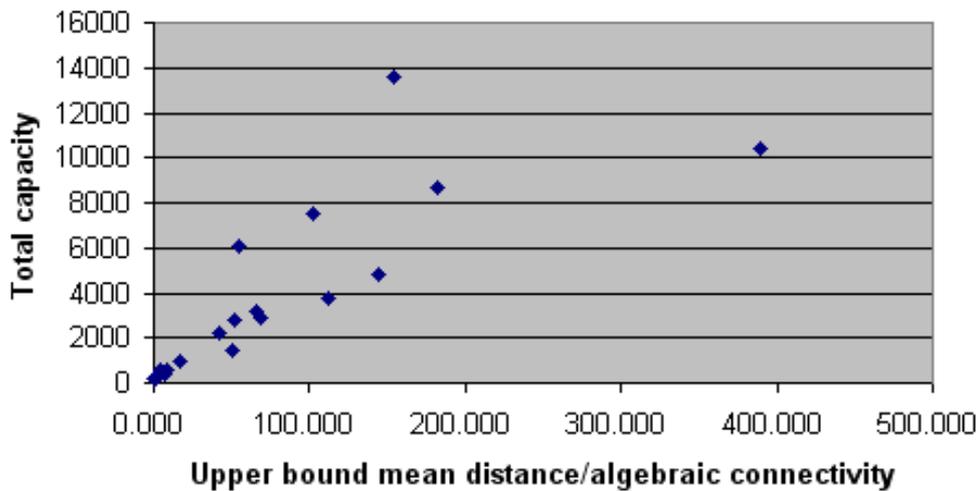


Figure 5.20 Total capacity vs. upper bound mean distance/algebraic connectivity

As shown in Table 5.7 and Figures 5.18 -5.20, it can be seen that, in Figure 5.18, there is no clear correlation between the total capacity and average nodal degree metric. For example, there are 7 network topologies have their average nodal degree about 3, but the total capacity solutions are significantly different and range from 431 units to 10412 units. As shown in Figures 19-20, we can find a monotonically increasing trend between the total capacity and mean distance related metric. In Figure 5.20, we found that, the total capacity

increases as the value of the combined metric  $\frac{\rho_{up}}{\lambda_2}$  increases. When  $\lambda_2$  is fixed, the total capacity increase as the  $\rho_{up}$  increases, since as the mean distance becomes larger, more working and protection capacity need to be allocated. In addition, as  $\rho_{up}$  is fixed, the total capacity decreases as the algebraic connectivity  $\lambda_2$  increases, since a larger  $\lambda_2$  indicates denser connectivity and thus more capacity sharing can be achieved. Moreover, we can see that the combined metric  $\frac{\rho_{up}}{\lambda_2}$ , which can be easily derived from the second smallest eigenvalue  $\lambda_2$ , have a monotonic trend as similar as the  $\frac{\rho_{exact}}{\lambda_2}$ , where  $\rho_{exact}$  is more complex to be calculated. Therefore, we can conclude that the algebraic connectivity metric,  $\lambda_2$ , not only can indicate the network connectivity, but also can be used to estimate the mean distance of a network in a convenient way.

## 5.6 Summary

We introduced an algebraic connectivity metric, adopted from spectral graph theory, namely the 2nd smallest eigenvalue of the Laplacian matrix of the network topology, as an alternative to the average nodal degree, to characterize network robustness in studies of the SCA problem. Extensive simulation studies confirmed that this metric is a more informative parameter than the average nodal degree for characterizing network topologies in survivability studies. In general, a larger algebraic connectivity means better network connectivity i.e., more node- and link-disjoint paths to choose from between node pairs, thus less network capacity would be allocated. Moreover, by considering the network mean distance characteristic, we can use a combined metric,  $\frac{\bar{\rho}}{\lambda_2}$ , to quantify different sizes of topologies and find a monotonically increasing trend between the total capacity and the  $\frac{\bar{\rho}}{\lambda_2}$  topological index.



# Chapter 6

## Conclusion and Future work

### 6.1 Contributions

The main target of this dissertation was to provide an in-depth understanding of the fundamentals of the spare capacity allocation (SCA) problem for survivable routing and to contribute new ideas and techniques to the area of network resiliency. There are three main contributions in this dissertation:

Firstly, the relationship between working capacity and spare capacity in SCA problem was addressed in Chapter 2. We introduced a matrix-based structure, called the Distributed Resilience Matrix (DRM) that can capture complete information on capacity usage in a distributed manner.

Secondly, we proposed a novel FoF-R ant-based algorithm to find the protection cycles and to explore the sharing potential among protection paths by introducing the Capacity Headroom-dependent Function (CHF), an attraction/repulsion relationship function. By using the proactive ant mobile agents to continuously investigate the network capacity usage, and update the protection cycle tables and DRMs at each node in the distributed control environment, our FoF-R algorithm showed good performance in trading off computational speed against capacity efficiency.

Finally, the relationship between spare capacity allocation and topological connectivity metrics such as average nodal degree vs. algebraic connectivity was addressed in Chapter 5. We showed that the average nodal degree of a network is not sufficient on its own for quantifying the network robustness. We suggested using a more informative metric: the algebraic connectivity of the network, as it provides a better numerical characterization of a

given topology and its dependence on key network connectivity properties. In general, a larger algebraic connectivity means better network connectivity i.e., more node- and link-disjoint paths exist that can be chosen from by pairs of communicating nodes, and so less network capacity needs to be allocated. It was also shown that there is a power law relationship between the total capacity and the algebraic connectivity metric.

## 6.2 Future work

Considering the work covered in this dissertation and the development of the future network, it would be useful to highlight some future areas of investigation.

The traffic scenario was assumed to be uniform for simplicity, hence more complex traffic scenarios should be used in future work.

Work on improving the effectiveness of the relationship function in the FoF-R algorithm should be explored. The algorithm should be improved or enhanced and verified in various other topologies and traffic scenarios. In addition, extension of the FoF-R routing algorithm to handle multiple failures and differentiated resilience requirements in a multi-service NGN environment would be a promising research direction.

More extensive studies on how the algebraic connectivity affects the amount of spare capacity to be allocated in more complex topologies are need, especially in different types of topology. Furthermore, capacitated versions of networks need to be studied, taking into account the fact that the network may have existing link capacities and/or link capacity limits to be respected with different traffic scenarios.

Further research work involving network topology design can investigate the following problems:

- **Generation:** can we efficiently generate ensembles of random but realistic topologies by replicating a set of simple graph metrics?
- **Evolution:** what are the forces driving the evolution (growth) of the topology of a given network?

## References

- [1] M.To and P.Neusy, "Unavailability analysis of long-haul networks," *IEEE Journal on Selection areas in Communications*, vol.12, no.1, January 1994, pp.100-109.
- [2] D.Crawford, "Fiber Optic Cable Dig-ups: causes and cures," in *Network Reliability and Interoperability Council website*, 1992.
- [3] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C.N.Chuah and C. Diot, "Characterization of Failures in an IP Backbone," in *Proc. of the INFOCOM 04*, pp.2307-2317 vol.4 Hong Kong, China, March 2004.
- [4] L. Nederlof, K. Struyve, C. O'Shea, H. Misser, D. Yonggang and B. Tamayo "End-to-end survivable broadband networks," *IEEE Communications Magazine*, pages 63–70, 9 1995.
- [5] S. Chen and K. Nahrstedt, "An overview of quality-of-service routing for the next generation high-speed networks problems and solutions," *IEEE Network Magazine*, Special Issues on Transmission and Distribution of Digital Video, vol.12, pp.64-79, November/December 1998.
- [6] P.-H. Ho and H. T. Mouftah, "A framework of service guaranteed shared protection for optical networks," *IEEE Communications Magazine*, pp. 97–103, February 2002.
- [7] P.-H. Ho, J. Tapolcai and H.T. Mouftah, "On optimal diverse routing for shared protection in mesh WDM networks," *IEEE Transactions on Reliability*, Vol.53, No.6, pp.216-225, June 2004.

- [8] G. Li, D. Wang, C. Kalmanek and R. Doverspike, "Efficient distributed path selection for shared restoration connections," in *IEEE Transactions on Networking*, vol. 11, pp. 761–771, October 2003.
- [9] Y. Xiong, D. Xu and C. Qiao, "Achieving fast and bandwidth efficient shared path protection," in *IEEE Journal of Lightwave Technology*, pp. 365–371, February 2003.
- [10] J. Zhou and X. Yuan, "A study of dynamic routing and wavelength assignment with imprecise network state information," in *Proc. of the International Conference on Parallel Processing Workshops, ICPPW'02*. Los Alamitos, CA, USA: IEEE Computer Society, 2002.
- [11] S. Shen, G. Xiao T.H. Cheng., "Benefits of Advertising Wavelength Availability in Distributed Lightpath Establishment," in *Proc. of the IEEE International Conference on Communications, ICC'05*, vol. 3, May 2005, pp. 1782–1786.
- [12] Y. Liu, D. Tipper and P. Siripongwutikorn, "Approximating optimal spare capacity allocation by successive survivable routing," in *Proc. of IEEE INFOCOM*, pp. 699–708, 2001.
- [13] G. Di Caro and M. Dorigo, "AntNet: A Mobile Agents Approach to Adaptive Routing," *Universite Libre de Bruxelles, Belgium*, 1997, Technical Report IRIDIA/97-12.
- [14] I. Kassabalidis, M. El-Sharkawi, R.j. Marks, P. Arabshahi and A.A. Gray, "Swarm Intelligence for Routing in Communication Networks," in *Proc. of the IEEE GLOBECOM'01*, San Antonio, Texas, November 2001, pp25–29.
- [15] S. Ramamurthy and B. Mukherjee, "Survivable WDM mesh networks, part I – protection," in *Proc. of IEEE INFOCOM*, New York, USA, March 21-25 1999. IEEE Press.
- [16] Y. Xiong and L. G. Maso, "Restoration strategies and spare capacity requirements in self-healing ATM networks," *IEEE/ACM Transactions on Networking*, 7(1):98–110, February 1999

- [17] W. D. Grover, "The selfhealing network: A fast distributed restoration technique for network using digital cross-connect machines," in Proc. of IEEE Globecom'87, Tokyo, pp.1090-1095.
- [18] W. D. Grover, "Distributed Restoration of the Transport Network," chapter 11 in Network Management into the 21<sup>st</sup> Century, IEEE/IEE Press co publication, 1994, pp.337-417
- [19] J. Wang, L. Sahasrabuddhe and B. Mukherjee, "Path vs. subpath vs. link restoration for fault management in IP-over-WDM networks: performance comparisons using GMPLS control signaling," Communications Magazine, IEEE Publication Date: Nov 2002 Volume: 40, Issue: 11 on page(s): 80- 87.
- [20] J. P. Lang and J. Drake, "Mesh Network Resiliency Using GMPLS," in Proc. of IEEE, vol. 90, no. 9, Sept. 2002, pp. 1559–64.
- [21] D. Xu, Y. Xiong, C. Qiao and G. Li, "Trap Avoidance and Protection Schemes in Networks with Shared Risk Link Groups," Journal of Lightwave Technology, vol. 20, no. 11, Nov. 2003, pp. 2683–93.
- [22] W. D. Grover and D. Stamatelakis, "Cycle-oriented Distributed Preconfiguration: Ring-like Speed with Mesh-like Capacity for Self-planning Network Restoration," in Proceedings of the IEEE International Conference on Communications (ICC), (Atlanta, GA, USA), June 1998, vol. 1, pp. 537–543.
- [23] J. Doucette, D. He, and W. D. Grover, "Algorithm approaches for efficient enumeration of candidate p cycle network design," Proc. Fourth International Workshop on the Design of Reliable Communication Networks (DRCN 2003), Banff, Alberta, Canada, vol. Oct., pp. 212–220, 2003.
- [24] D. Stamatelakis and W. D. Grover, "Theoretical Underpinnings for the Efficiency of Restorable Networks Using Pre-configured Cycles ("p-cycles")," IEEE Transactions on Communications, vol.48, no.8, pp. 1262-1265, August 2000.

[25] D. Stamatelakis and W. D. Grover, "IP Layer Restoration and Network Planning Based on Virtual Protection Cycles," *IEEE Journal on Selected Areas in Communications*, Special Issue on Protocols and Architectures for Next Generation Optical WDM Networks, vol.18, no.10, pp. 1938-1949, October, 2000.

[26] W. D. Grover, *Mesh-Based Survivable Networks: Options and Strategies for Optical, MPLS, SONET, and ATM Networking*, Prentice Hall PTR, Upper Saddle River, NJ, 2004.

[27] W. D. Grover, J. Doucette, M. Clouqueur, D. Leung and D. Stamatelakis, "New options and insights for survivable transport networks," *IEEE Communications Magazine*, 40(1):34–41, 2002.

[28] W.-P. Wang, D. Tipper, B. Jager and D. Medhi, "Fault recovery routing in wide area packet networks," in *Proc. of 15th International Teletraffic Congress*, Washington, DC, June 1997.

[29] R. Cotter, D. Medhi, and D. Tipper, "Traffic backlog and impact on network dimensioning for survivability for wide-area VP-based ATM networks," in *Proc. of 15<sup>th</sup> International Teletraffic Congress*, Washington, DC, June 1997.

[30] P. Soriano, E. Comerciales, C. Wynants and M. Gendreau, "Design and dimensioning of survivable SDH / SONET Networks," *Telecommunications Network Planning*, eds. B. Sanso, P. Soriano, Kluwer Academic, pp. 148-167, 1999.

[31] H. Sakauchi, Y. Nishimura and S. Hasegawa, "A Self-Healing Network with an Economical Spare Channel Assignment," in *Proc of IEEE Global Telecommunications Conference (GlobeCom 1990)*, San Diego, CA, vol. 1, pp. 438-441, December 1990.

[32] B. D. Venables, W. Grover, and M. H. MacGregor, "Two Strategies for Spare Capacity Placement (SCP) in Mesh Restorable Networks," in *Proc. of IEEE International Conference on Communications (ICC 1993)*, pp. 267-271, Geneva, Switzerland, May 1993.

- [33] M. Herzberg and S. J. Bye, "An optimal spare-capacity assignment model for survivable networks with hop limits," in Proc. of IEEE Global Communications Conference (GlobeCom 1994), pp. 1601-1607, San Francisco, CA, December 1994.
- [34] M. Herzberg, S. J. Bye and A. Utano, "The hop-limit approach for spare-capacity assignment in survivable networks," IEEE/ACM Transactions on Networking, 3(6):775–784, December 1995.
- [35] W. L. Winston, Operations Research Applications and Algorithms, 3<sup>rd</sup> Edition, Duxbury Press, Belmont, CA, 1994.33, June 1998.
- [36] M. Grotchel, C. Monma and M.Stoer, "Polyhedral and computational investigations for designing communication networks with high survivability requirements," Operations Research, 43(6):1012-1024, 1995. 29.
- [37] S. Soni, R. Gupta and H.Pirkul, "Survivable network design: The state of the art," Information Systems Frontiers, 1(3):303-315, 1999.
- [38] B. Fortz, M Labbe and F.Maffioli, "Solving the two-connected network with bounded meshes problem," Operations Research, 48:866-877, 2000.
- [39] Y. Wang, Modeling and solving single and multiple facility restoration problems, Ph.D. dissertation, Sloan School of Management, Massachusetts Institute of Technology 1998.
- [40] R. Ahuja, T. Magnanti, J. Orlin, Network Flows: Theory, Algorithms, and Applications, Englewood Cliffs, New Jersey, Prentice Hall, 1993.
- [41] B.H. Ryu, M.Murata and H.Miyahara, "Design method for highly reliable virtual path based ATM networks," IEICE Transactions on Communications, E79-B(10):1500–1514, 10 1996.
- [42] M. Kodialam and T.V. Lakshman, "Dynamic routing of bandwidth guaranteed tunnels with restoration," in Proc. of IEEE INFOCOM, March 2000

- [43] X. Su and C.-F. Su, "An online distributed protection algorithm in WDM networks," in Proc. of IEEE International Conference on Communications, pages 1571–1575, 2001.
- [44] W. D. Grover and D. Y. Li, "The forcer concept and express route planning in mesh-survivable networks," Journal of Network and System Management, 7(2):199–223, 1999
- [45] W.D. Grover, V. Rawat and M.H.MacGregor, "Fast heuristic principle for spare capacity placement in mesh- restorable SONET/SDH transport networks," Electronics Letters, 33(3):195–196, Jan 1997.
- [46] C.Qiao and D. Xu, "Distributed partial information management (DPIM) schemes for survivable networks - part I," in Tech Report 2000-13, CSE Dept. University at Buffalo.
- [47] D.Xu and C.Qiao, "Distributed partial information management (DPIM) schemes for survivable networks - part II," Tech Report 2000-14, CSE Dept., University at Buffalo.
- [48] M. Kodialam and T.V. Lakshman, "Dynamic Routing of Locally Restorable Bandwidth Guaranteed Tunnels Using Aggregated Link Usage Information," in Proc. IEEE INFOCOM '01, pp. 376-385, 2001.
- [49] C. Dovrolis and P. Ramanathan, "Resource aggregation for fault tolerance in integrated service networks," ACM Computer Communication Review, 28(2):39–53, 1998.
- [50] S. Cwilich, M. Deng, D. F. Lynch, S. J. Phillipsy and J. R. Westbrooky, "Algorithms for restoration planning in a telecommunications network, " in Algorithm Engineering and Experimentation, Intl. Workshop, ALENEX'99, Lecture Notes in Computer Science 1619, volume 1619, pages 194–209, 1999
- [51] J. Duato, "A theory of fault-tolerant routing in wormhole networks," IEEE Trans. on Parallel and Distributed Systems, 8(8):790–802, August 1997.

- [52] S. Norden, M. M. Buddhikot, M. Waldvogel and S. Suri, "Routing bandwidth guaranteed paths with restoration in label switched networks," *Network Protocols*, 2001. Ninth International Conference on Volume, Issue, 11-14 Nov. 2001 Page(s): 71 – 79.
- [53] Y. Liu, D. Tipper and P. Siripongwutikorn, "Approximating Optimal Spare Capacity Allocation by Successive Survivable Routing," in *Proc. of IEEE INFOCOM '01*, pp. 699-708, 2001.
- [54] Y. Liu and D. Tipper, "Successive survivable routing for node failures," in *IEEE GLOBECOM 2001*, San Antonio, TX, Nov. 25-29, 2001.
- [55] Y. Liu, D. Tipper, and P. Siripongwutikorn, "Approximating optimal spare capacity allocation by successive survivable routing," *IEEE/ACM Transaction on Networking*, vol. 13, no. 1, Feb. 2005, pp. 198-211
- [56] Y.Liu, *Spare Capacity Allocation: Method, Analysis and Algorithm*, PhD. thesis
- [57] J. Tapolcai, P-H. Ho and A.Haque, "TROP: A Novel Approximate Link-State Dissemination Framework For Dynamic Survivable Routing in MPLS Networks," *IEEE Trans. on Parallel and Distributed Systems*, Volume 19, Issue 3, March 2008 Page(s):311 – 322
- [58] J. Tapolcai, P-H. Ho, X. Jiang, and S. Horiguchi, " Shared Protection Based on Matrix Decomposition in Tropical Semi-Rings, " in *Proc. of AINA 2005*: 655-660
- [59] J. Tapolcai, P.-H. Ho, X. Jiang, and S. Horiguchi, "A Study on Distributed Control Architectures for Shared Protection," *Proc.IEEE GLOBECOM*, 2004.
- [60] L. Ruan and H. Luo, "Dynamic Routing of Restorable Lightpaths: A Tradeoff between Capacity Efficiency and Resource Information Requirement," *Proc. Seventh IFIP Working Conf. Optical Network Design and Modelling (ONDM '03)*, pp. 537-548, 2003.

- [61] R. Schoonderwoerd, O. Holland, J. Bruten, L. Rothkrantz, "Ant-based Load Balancing in Telecommunications Networks," *Adaptive Behavior*, 5(2):169–207, 1997.
- [62] G. Di Caro and M. Dorigo, "AntNet: Distributed Stigmergetic Control for Communications Networks," *Journal of Artificial Intelligence Research*, 9:317–365, December 1998.
- [63] T. White, B. Pagurek, F. Oppacher, "Connection Management using Adaptive Mobile Agents," *Proc.s of 1998 International Conference on Parallel and Distributed Processing Techniques and Applications (PDAPTA'98)*, 1998.
- [64] J. Deneubourg and J. Gross, "Collective Patterns in Decision Making," in *Ethology Ecology and Evolution*, 1989.
- [65] E. Bonabeau, G. Theraulza, J.-L. Deneubourg, S. Aron, and S. Camazine, "Self-Organization in Social Insects," *Trends in Ecol. and Evol.*, vol. 12, no. 5, pp. 188–193, 1997.
- [66] E. Bonabeau, M. Dorigo, G. Theraulaz, "Swarm Intelligence: From Natural to Artificial Systems," Oxford University Press, 1999.
- [67] R. Beckers, J. Deneubourg, S. Goss, and J.-M. Pasteels, "Collective Decision Making Through Food Recruitment," *Insect Societies*, vol. 37, pp. 258–267, 1990.
- [68] A. Colourni, M. Dorigo, and V. Maniezzo, "Distributed Optimization by Ant Colonies," in *Toward a Practice of Autonomous Systems: Proc.s of The First European Conference on Artificial Life*. The MIT Press, 1992, pp. 134–142.
- [69] M. Dorigo and V. Maniezzo, "Ant System: Optimization by a Colony of Cooperating Agents," *IEEE Trans. Syst, Man, Cybernetics, Part B*, vol. 26, no. 1, pp. 29–41, Feb. 1996.
- [70] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy-Kan, and D. B. Shmoys, *The Travelling Salesman Problem*. Wiley, 1985.

- [71] T. C. Koopmans and M. J. Berkmann., “Assignment Problems and the Location of Economic Activities. *Econometrica*,” *Econometrica*, vol. 25, pp. 53–76, 1957.
- [72] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by Simulated Annealing,” *Science*, vol. 220, no. 4598, pp. 671–680, May 1983.
- [73] B. Bullnheimer, R. Hartl, and C. Strauss, “Applying the Ant System to the Vehicle Routing Problem,” *Sophia-Antipolis, France, 1997*, Presented at the 2nd Metaheuristic International Conference.
- [74] Q. Costa and A. Hertz, “Ants Can Colour Graphs,” *Journal of the Operations Research Society*, vol. 48, pp. 295–305, 1997.
- [75] G. Bilchev and I. C. Parmee, “The Ant Colony Metaphor for Searching Continuous Design Spaces,” in *Selected Papers from AISB Workshop on Evolutionary Computing*. London, UK: Springer-Verlag, 1995, pp. 25–39.
- [76] M. Dorigo and T. Stutzle, *Ant Colony Optimization*, Bradford Book, 2004.
- [77] D. Bertsekas and R. Gallager, *Data networks*. Upper Saddle River, NJ, USA:Prentice-Hall, Inc., 1987.
- [78] A. Bieszczad, T. White, and B. Pagurek, “Mobile Agents for Network Management,” *IEEE Communications Surveys*, vol. 1, no. 1, pp. 5–9, 1998.
- [79] N. Minar, K. H. Kramer, and P. Maes, “Cooperating Mobile Agents for Dynamic Network Routing,” in *Software Agents for Future Communication Systems*. Springer-Verlag: Heidelberg, Germany, 1999, pp. 287–304.
- [80] D. S. Milojicic, “Trend Wars: Mobile Agent Applications,” *IEEE Concurrency*, vol. 7, no. 3, pp. 80–90, 1999.

- [81] G. Di Caro and M. Dorigo, "Mobile Agents for Adaptive Routing," in Proc. of the Thirty-First Annual Hawaii International Conference on System Sciences, HICSS'98. Washington, DC, USA: IEEE Computer Society, 1998, p. 74.
- [82] E. Bonabeau, M. Dorigo, and G. Theraulaz, *From Natural to Artificial Swarm Intelligence*. Oxford University Press, 1999.
- [83] S. Marwaha, C. K. Tham, and D. Srinivasan, "Mobile Agents Based Routing Protocol for Mobile Ad Hoc Networks," in Proc. of the GLOBECOM'02, vol. 1, New York, 2002, pp. 163–167.
- [84] M. Gunes, U. Sorges, and I. Bouazisi, "ARA - The Ant-Colony Based Routing Algorithm for MANETs," in Proc. of the 2002 International Conference on Parallel Processing Workshops, ICPPW'02. Washington, DC, USA: IEEE Computer Society, 2002, pp. 79–85. 106
- [85] O. Hussein and T. Saadawi, "Ant Routing Algorithm for Mobile Ad-hoc Networks," in Proc. of the 22nd IEEE International Performance, Computing, and Communications Conference, IPCCC'03, Phoenix, Arizona, Apr. 2003, pp. 281–290.
- [86] C.-C. Shen, C. Jaikao, C. Srisathapornphat, Z. Huang, and S. Rajagopalan, "Ad hoc Networking with Swarm Intelligence," in Proc. of the Third International Workshop on Ant Algorithms, ANTS'04, ser. LNCS 3172. Springer-Verlag, 2004, pp. 262–269.
- [87] G. Di Caro, F. Ducatelle, and L. M. Gambardella, "AntHocNet: An Ant-Based Hybrid Routing Algorithm for Mobile Ad Hoc Networks," in Proc. of Parallel Problem Solving from Nature, PPSN VIII, ser. LNCS 3242. Springer-Verlag, 2004, pp. 461–470.
- [88] O. Hussein, T. Saadawi, and M. Lee, "Probability Routing Algorithm for Mobile Ad hoc Networks," *IEEE J. Select. Areas Commun.*, vol. 23, no. 12, pp. 2248–2259, Dec. 2005.
- [89] R. Schoonderwoerd, O. Holland, J. Bruten, L. Rothkrantz, "Ant-based Load Balancing in Telecommunications Networks," *Adaptive Behavior*, 5(2):169–207, 1997.

- [90] G. Di Caro, M. Dorigo, “AntNet: Distributed Stigmergetic Control for Communications Networks,” *Journal of Artificial Intelligence Research*, 9:317–365, December 1998.
- [91] K. M. Sim and W. H. Sun, “Multiple ant-colony optimization for network routing,” in *Proc. 1st Int. Symp. Cyberworld*, Tokyo, Japan, November 2002, pp. 277–281.
- [92] K. M. Sim and W. H. Sun, “A multiple ant colony optimization approach for load balancing,” in *Proc. 4th Int. Conf. Intelligent Data Engineering Automated Learning*, Hong Kong, 2003.
- [93] N. Varela and M. C. Sinclair, “Ant colony optimization for virtual-wavelength- path routing and wavelength allocation,” in *Proc. Congress Evolutionary Computation*, Washington, DC, July 1999, pp. 1809–1816.
- [94] S. Fenet and S. Hassas, “An ant system for multiple criteria balancing,” in *Proc. 1st Int. Workshop Ants Systems*, Brussels, Belgium, Sept 1998
- [95] A. Nowe., K. Verbeeck and P. Vranckx, “Multi-Type Ant Colony System: the Edge disjoint path problem,” *Proc the ANTS 2004 Workshop*, LNCS vol 3172, p202-213, 2004.
- [96] O. Wittner and B. E. Helvik, “Cross Entropy Guided Ant-like Agents Finding Dependable Primary/Backup Path Patterns in Networks,” *Proc. CEC2002*, Honolulu, Hawaii, May 12-17th 2002.
- [97] OMNeT++ website: <http://www.omnetpp.org/>
- [98] R. Fourer, D. M. Gay, and B. W. Kernighan, *AMPL: A Modeling Language for Mathematical Programming*. San Francisco, CA, 1993.
- [99] *CPLEX User Manual v11.1*, ILOG, Inc., 2008.
- [100] SNDlib: <http://sndlib.zib.de/home.action>

[101] B. Mohar, "Some applications of Laplace eigenvalues of graphs," *Graph Symmetry: Algebraic Methods and Applications*, volume 497 of NATO ASI Series C, 1997, pages 227-275.

[102] B. Forst and W. D. Grover, "Factors affecting the efficiency of Demand-wise Shared Protection, " in *Proc. of workshop on the Design of Reliable Communication Networks (DRCN 2007)*, On page(s): 1-8.

[103] B. Todd and J. Doucette, "Multi-Flow Optimization Model for Design of a Shared Backup Path Protected Network, " in *Proc. of the ICC '08. IEEE International Conference on Communications*, 19-23 May 2008, page(s): 131 -138.

[104] J. Doucette and W. D. Grover, "Comparison of mesh protection and restoration schemes and the dependency on graph connectivity, " in *Proc. of 3 rd Int. Workshop on the Design of Reliable Communication Networks (DRCN 2001)*, pp.121-128.

[105] W. D. Grover and J. Doucette, "Increasing the Efficiency of Span-restorable Mesh Networks on Low-connectivity Graphs, " in *Proc. of 3 rd Int. Workshop on the Design of Reliable Communication Networks (DRCN 2001)*, pp.99-106.

[106] F.R.K. Chung, *Spectral Graph Theory*, CBMS Regional Conference Series in Mathematics, No. 92, 1997

[107] M. Fiedler, "Algebraic connectivity of graphs," *Czechoslovak Mathematical J.*, vol. 23, pp. 298–305, 1973.

[108] M. Fiedler, "A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory, " *Czechoslovak Mathematical J.*, vol. 25, pp. 619–633, 1975

[109] N.Alon , Eigenvalues and expanders. *Combinatorica*, 6(2):83-96,1986

[110] M. Jerrum and A. Sinclair , “Conductance and the rapid mixing property for Markov chains: the approximation of permanent resolved,” in Proc. of the twentieth annual ACM symposium on Theory of computing, p.235-244, May 02-04, 1988, Chicago, Illinois, United States

[111] M. Mihai, “Conductance and convergence of Markov chains (A combinatorial Treatmeat of expanders),” FOCS 1989, pp 526-531.

[112] Bojan Mohar, “Eigenvalues, Diameter and Mean Distance in graphs,” Graphs Combin. 7 (1991), pp. 53–64.