

1-1-2012

Optimal Capacity And Flow Assignment For Self-Healing ATM Networks Based On Path Restoration: Nonlinear Models

Faria Khandaker
Ryerson University

Follow this and additional works at: <http://digitalcommons.ryerson.ca/dissertations>



Part of the [Computer Sciences Commons](#)

Recommended Citation

Khandaker, Faria, "Optimal Capacity And Flow Assignment For Self-Healing ATM Networks Based On Path Restoration: Nonlinear Models" (2012). *Theses and dissertations*. Paper 1529.

**OPTIMAL CAPACITY AND FLOW ASSIGNMENT FOR SELF-HEALING ATM
NETWORKS BASED ON PATH RESTORATION: NONLINEAR MODELS**

by

Faria Khandaker

M.Sc., University of Dhaka, Bangladesh, 2007

A Thesis

presented to Ryerson University

in partial fulfillment of the
requirements for the degree of

Master of Science

in the Program of

Computer Science

Toronto, Ontario, Canada, 2012

© Faria Khandaker, 2012

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my thesis may be made electronically available to the public.

Abstract

Optimal Capacity and Flow Assignment for Self-Healing ATM Networks Based on Path

Restoration: Nonlinear Models

© Faria Khandaker, 2012

Master of Science

Computer Science

Ryerson University

This thesis addresses the design of self-healing Asynchronous Transfer Mode (ATM) networks which is a special aspect of a more general problem, referred to as capacity and flow assignment (CFA) problem in self-healing ATM networks. We have proposed two nonlinear mathematical models for global reconfiguration strategy and failure-oriented reconfiguration strategy in our thesis. Our restoration strategies aim to minimize the capacity installation cost and the routing cost when a single link failure occurs in the network. A special case of the augmented Lagrangian method so-called Separable Augmented Lagrangian Algorithm (SALA) is proposed for solving the proposed nonlinear mathematical models. Numerical results are presented comparing the two restoration strategies in terms of five performance metrics which are capacity installation cost, total required capacity, routing cost, total network cost and required CPU time for convergence of the algorithms. Our results show that the global reconfiguration strategy has always performed better than the failure-oriented reconfiguration strategy for all the network scenarios, topologies and bandwidth requirements.

Acknowledgement

I would first like to thank my supervisor, Dr. Isaac Woungang, and my co-supervisor Dr. Oumar Mandione Guèye, for their support, patience and time throughout my graduate studies. I am finally able to achieve my goal for their strong support and help. They did not only guide me in the course of my thesis, but also provided me an opportunity to be benefited from their vast knowledge. They were always present to help me out and to guide me whenever I needed their guidance.

I would like to acknowledge my gratitude to the Department of Computer Science and the School of Graduate Studies of Ryerson University, for their financial supports and all the opportunities.

I would like to thank my parents for their support, love and advice. Without them, I would not be here today. Finally, and most importantly, I would like to thank my husband A.B.M. Bodrul Alam, who has been a constant source of encouragement for me throughout my studies. Without his support, courage and adorableness, I would not be able to achieve my goal.

Dedication

To my husband

A.B.M. Bodrul Alam

I would not be here without his love, support and strong belief in me.

Contents

1 Introduction	1
1.1 Context of Our Study	1
1.2 Motivation	2
1.3 Research Problem.....	3
1.4 Objectives of The Thesis.....	3
1.5 Research Scope	4
1.6 Organization of The Thesis	4
2 Background and Literature Review	6
2.1 Background	6
2.1.1 Asynchronous Transfer Mode Protocol.....	6
2.1.1.1 ATM Network Architecture	8
2.1.1.2 Routing in ATM Network	11
2.1.1.3 Current Transport Control Plane	11
2.1.2 Self-Healing Networks	12
2.1.3 Network Restoration.....	12
2.1.3.1 Link Restoration	14
2.1.3.2 Path Restoration.....	15
2.1.4 Mathematical Optimization Techniques	17
2.1.4.1 Linear Optimization Problems.....	17
2.1.4.2 Nonlinear Optimization Problems	18
2.1.4.3 Unconstrained Nonlinear Optimization Problems.....	18
2.1.4.4 Constrained Nonlinear Optimization Problems.....	20
2.1.4.5 Penalty Methods	20
2.1.4.6 Lagrangian Method.....	21
2.1.4.7 Augmented Lagrangian Methods	22
2.2 Literature Review	23
3 Optimization Models and Solution Approaches	28
3.1 Mathematical Model for the Global Reconfiguration Strategy.....	28
3.2 Mathematical Model for the Failure-Oriented Reconfiguration Strategy.....	30

3.3 Solution Approaches	33
3.3.1 Separable Augmented Lagrangian Algorithm (SALA).....	34
3.3.2 Dijkstra Algorithm.....	36
3.3.3 Modified Newton Method	37
3.3.4 Armijo Method	38
3.4 SALA Applied to the Global Reconfiguration Strategy	39
3.5 SALA Applied to the Failure-Oriented Reconfiguration Strategy.....	43
4 Performance Evaluation.....	49
4.1 Parameters for Global and Failure-Oriented Reconfiguration Strategies	49
4.1.1 Network Topologies	49
4.1.2 Traffic Demands.....	50
4.1.3 Network Scenarios.....	50
4.1.4 Performance Metrics.....	51
4.1.4.1 Capacity Installation Cost.....	51
4.1.4.2 Total Used Capacity	51
4.1.4.3 Routing Cost.....	51
4.1.4.4 Total Network Cost	52
4.1.4.5 CPU Time.....	52
4.2 Comparison of Global and Failure-Oriented Reconfiguration Strategies.....	54
4.2.1 Results Using Scenario 1	54
4.2.1.1 Capacity Installation Cost.....	54
4.2.1.2 Total Used Capacity	57
4.2.1.3 Routing Cost.....	59
4.2.1.4 Total Network Cost	62
4.2.1.5 CPU Time	64
4.2.2 Results Using Scenario 2.....	66
4.2.2.1 Capacity Installation Cost.....	66
4.2.2.2 Total Used Capacity	69
4.2.2.3 Routing Cost.....	72
4.2.2.4 Total Network Cost	74
4.2.2.5 CPU Time.....	77

4.2.3 Results Using Scenario 3.....	79
4.2.3.1 Capacity Installation Cost.....	79
4.2.3.2 Total Used Capacity	80
4.2.3.3 Routing Cost.....	82
4.2.3.4 Total Network Cost	83
4.2.3.5 CPU Time.....	84
5 Conclusion	86
Appendix A	88
References.....	94

List of Tables

3.1: Notations used in the mathematical model for Global Reconfoguration Strategy	28
3.2: Notations used in the mathematical model for Failure-Oriented Reconfoguration Strategy	30
4.1: Summary of all parameters for Global and Failure-Oriented Reconfiguration Strategies	52
4.2: Capacity Installation Cost (in Dollars) for Global vs. Failure-Oriented Reconfiguration Strategy using symmetric traffic demands.....	54
4.3: Capacity Installation Cost (in Dollars) for Global vs. Failure-Oriented Reconfiguration Strategy using asymmetric traffic demands.....	56
4.4: Total Used Capacity (in Bits) for Global vs. Failure-Oriented Reconfiguration Strategy using symmetric traffic demands.....	57
4.5: Total Used Capacity (in Bits) for Global vs. Failure-Oriented Reconfiguration Strategy using asymmetric traffic demands.....	58
4.6: Routing Cost (in Dollars) for Global vs. Failure-Oriented Reconfiguration Strategy using symmetric traffic demands.....	59
4.7: Routing Cost (in Dollars) for Global vs. Failure-Oriented Reconfiguration Strategy using asymmetric traffic demands.....	61
4.8: Total Network Cost (in Dollars) for Global vs. Failure-Oriented Reconfiguration Strategy using symmetric traffic demands	62
4.9: Total Network Cost (in Dollars) for Global vs. Failure-Oriented Reconfiguration Strategy using asymmetric traffic demands	63
4.10: CPU Time (in Seconds) for Global vs. Failure-Oriented Reconfiguration Strategy using symmetric traffic demands.....	65
4.11: CPU Time (in Seconds) for Global vs. Failure-Oriented Reconfiguration Strategy using asymmetric traffic demands.....	65
4.12: Capacity Installation Cost (in Dollars) for Global vs. Failure-Oriented Reconfiguration Strategy using symmetric traffic demands.....	67
4.13: Capacity Installation Cost (in Dollars) for Global vs. Failure-Oriented Reconfiguration Strategy using asymmetric traffic demands.....	68
4.14: Total Used Capacity (in Bits) for Global vs. Failure-Oriented Reconfiguration Strategy using symmetric traffic demands.....	70

4.15: Total Used Capacity (in Bits) for Global vs. Failure-Oriented Reconfiguration Strategy using asymmetric traffic demands	71
4.16: Routing Cost (in Dollars) for Global vs. Failure-Oriented Reconfiguration Strategy using symmetric traffic demands.....	72
4.17: Routing Cost (in Dollars) for Global vs. Failure-Oriented Reconfiguration Strategy using asymmetric traffic demands.....	73
4.18: Total Network Cost (in Dollars) for Global vs. Failure-Oriented Reconfiguration Strategy using symmetric traffic demands	74
4.19: Total Network Cost (in Dollars) for Global vs. Failure-Oriented Reconfiguration Strategy using asymmetric traffic demands	76
4.20: CPU Time (in Seconds) for Global vs. Failure-Oriented Reconfiguration Strategy using symmetric traffic demands.....	77
4.21: CPU Time (in Seconds) for Global vs. Failure-Oriented Reconfiguration Strategy using asymmetric traffic demands.....	78
4.22: Capacity Installation Cost (in Dollars) for Global vs. Failure-Oriented Reconfiguration Strategy in different traffic demands	79
4.23: Total Used Capacity (in Bits) for Global vs. Failure-Oriented Reconfiguration Strategy in different traffic demands.....	81
4.24: Routing Cost (in Dollars) for Global vs. Failure-Oriented Reconfiguration Strategy in different traffic demands.....	82
4.25: Total Network Cost (in Dollars) for Global vs. Failure-Oriented Reconfiguration Strategy in different traffic demands.....	83
4.26: CPU Time (in Seconds) for Global vs. Failure-Oriented Reconfiguration Strategy in different traffic demands.....	85

List of Figures

2.1: An ATM cell.....	6
2.2: ATM Multiplexing.....	7
2.3: Relationship among the Transmission path, Virtual paths and Virtual circuits	7
2.4: Connection identifiers in a virtual circuit of ATM network.....	8
2.5: Architecture of an ATM Network	9
2.6: ATM layers	9
2.7: Routing in ATM network with a switch	11
2.8: Link Restoration Strategy	14
2.9: Path Restoration Strategy.....	15
2.10: Categories of Restoration Schemes for a Self-healing Network	16
4.1: Capacity Installation Cost for Global vs. Failure-Oriented Reconfiguration Strategy using symmetric traffic demands.....	55
4.2: Capacity Installation Cost for Global vs. Failure-Oriented Reconfiguration Strategy using asymmetric traffic demands.....	56
4.3: Total Used Capacity for Global vs. Failure-Oriented Reconfiguration Strategy using symmetric traffic demands.....	57
4.4: Total Used Capacity for Global vs. Failure-Oriented Reconfiguration Strategy using asymmetric traffic demands.....	59
4.5: Routing Cost for Global vs. Failure-Oriented Reconfiguration Strategy using symmetric traffic demands.....	60
4.6: Routing Cost for Global vs. Failure-Oriented Reconfiguration Strategy using asymmetric traffic demands.....	61
4.7: Total Network Cost for Global vs. Failure-Oriented Reconfiguration Strategy using symmetric traffic demands	62
4.8: Total Network Cost for Global vs. Failure-Oriented Reconfiguration Strategy using asymmetric traffic demands	64
4.9: CPU Time for Global vs. Failure-Oriented Reconfiguration Strategy using symmetric traffic demands.....	65
4.10: CPU Time for Global vs. Failure-Oriented Reconfiguration Strategy using	

asymmetric traffic demands	66
4.11: Capacity Installation Cost for Global vs. Failure-Oriented Reconfiguration Strategy using symmetric traffic demands	67
4.12: Capacity Installation Cost for Global vs. Failure-Oriented Reconfiguration Strategy using asymmetric traffic demands	69
4.13: Total Used Capacity for Global vs. Failure-Oriented Reconfiguration Strategy using symmetric traffic demands	70
4.14: Total Used Capacity for Global vs. Failure-Oriented Reconfiguration Strategy using asymmetric traffic demands	71
4.15: Routing Cost for Global vs. Failure-Oriented Reconfiguration Strategy using symmetric traffic demands	72
4.16: Routing Cost for Global vs. Failure-Oriented Reconfiguration Strategy using asymmetric traffic demands	74
4.17: Total Network Cost for Global vs. Failure-Oriented Reconfiguration Strategy using symmetric traffic demands	75
4.18: Total Network Cost for Global vs. Failure-Oriented Reconfiguration Strategy using asymmetric traffic demands	76
4.19: CPU Time for Global vs. Failure-Oriented Reconfiguration Strategy using symmetric traffic demands	77
4.20: CPU Time for Global vs. Failure-Oriented Reconfiguration Strategy using asymmetric traffic demands	78
4.21: Capacity Installation Cost for Global vs. Failure-Oriented Reconfiguration Strategy in different traffic demands	80
4.22: Total Used Capacity for Global vs. Failure-Oriented Reconfiguration Strategy in different traffic demands	81
4.23: Routing Cost for Global vs. Failure-Oriented Reconfiguration Strategy in different traffic demands	82
4.24: Total Network Cost for Global vs. Failure-Oriented Reconfiguration Strategy in different traffic demands	84
4.25: CPU Time for Global vs. Failure-Oriented Reconfiguration Strategy in different traffic demands	85

List of Abbreviations

CFA	Capacity and Flow Assignment
ATM	Asynchronous Transfer Mode
SALA	Separable Augmented Lagrangian Algorithm
QoS	Quality of Service
GMPLS	Generalized Multi-protocol Label Switching
TP	Transmission Path
VP	Virtual Path
VC	Virtual Circuit
VPI	Virtual Path Identifier
VCI	Virtual Circuit Identifier

Chapter 1

Introduction

1.1 Context of Our Study

This thesis deals with the design of a self-healing Asynchronous Transfer Mode (ATM) network which is a special aspect of a more general problem, referred to as capacity and flow assignment (CFA) problem in self-healing ATM networks. We have proposed, implemented, and evaluated two nonlinear restoration strategies, namely global reconfiguration strategy and failure-oriented reconfiguration strategy for designing a self-healing ATM network. The main objective of our proposals is to minimize the total network cost while fulfilling all traffic constraints when a single link failure occurs in the network.

In today's society, business, political, and social activities happen in a fast manner. Information needs to be sent across different parties of these activities almost instantly. In view of this, the ATM network has played a major role in both local area networks (LANs) and wide area networks (WANs). This is because ATM network outshines older generation network technologies by being able to provide larger bandwidth, different quality of services (QoS), guaranteed throughputs, as well as bounded delays for different kinds of traffic, including data, voice, video and even mixtures of these. These points make the ATM one of the best communication technologies for certain types of telecommunication applications in today's world. However, the more heavily a network is relied on, the more disastrous it becomes if failures happen to it [1]. So, it is very crucial to design survivable ATM networks.

Network survivability has become a critical issue in telecommunication networks due to increasing societal dependence on communication systems and the growing importance of information. Fast restoration from a network failure has been recognized as a key ingredient in realizing survivable networks in emerging high-speed ATM environments [2]. Different network survivability techniques have been proposed to guarantee seamless communication services for ATM networks when a failure occurs in the network. ATM networks with fault tolerance capability are called “self-healing ATM networks” [3].

In case of network survivability, the design of self-healing ATM networks is a well known problem named as the capacity and flow assignment (CFA) problem [3]. When the network topology and traffic demands of the network are known, the CFA problem consists of determining the optimal capacity and optimal traffic flow with minimal cost that meet the traffic demands even if a failure occurs in the network. Our research focuses on the CFA problem of ATM networks and proposes two restoration strategies for designing self-healing ATM networks.

1.2 Motivation

In the literature of network survivability, a lot of attention has been paid in the design of self-healing ATM networks. But this is often formulated as a linear multicommodity flow problem [3-5]. To the best of our knowledge, only few research works have treated the design of self-healing ATM networks as a nonlinear multicommodity flow problem.

The nonlinearity property comes from the objective function of the self-healing network design. The linear objective function of the self-healing ATM network design aims to minimize only the spare capacity installation cost where the working capacity of the network is fixed. But

our motivation is to minimize the total network cost, here considered as the aggregation of the capacity installation cost and routing cost. Routing cost is a very important quality of service (QoS) criterion in self-healing network design. The linear multicommodity flow problem is much easier to solve than the nonlinear case. In this thesis, we have taken the challenge to address the CFA problem as a nonlinear programming model; introducing a novel Lagrangian based approach, so-called Separable Augmented Lagrangian Algorithm (SALA) for solving it.

1.3 Research Problem

Our research addresses the design of self-healing ATM networks under the path restoration strategy. We have proposed nonlinear mathematical models for path restoration strategy for self-healing ATM networks. The proposed mathematical models can be considered as a special case of the CFA problem taking into account the restoration strategies. More precisely, our research objective is to propose and implement a global reconfiguration strategy and a failure-oriented reconfiguration strategy for self-healing ATM networks assuming the network topology and the traffic demands of the network are given.

1.4 Objectives of the Thesis

The objectives of this thesis can be summarized as follows:

1. Formulating the CFA problem as a nonlinear programming model using global reconfiguration strategy and failure-oriented reconfiguration strategy.
2. Proposing a special case of the augmented Lagrangian approach called Separable Augmented Lagrangian Algorithm (SALA) for solving the above mentioned nonlinear programming problems.

3. Comparing the global reconfiguration strategy and the failure-oriented reconfiguration strategy quantitatively in terms of several performance metrics.

1.5 Research Scope

In telecommunication networks, the probability of two network components failing simultaneously is very small; designing a network protected against single component failure is considered satisfactory [6]. Hence, our proposed global and failure-oriented reconfiguration strategies reroute all affected traffics of the ATM network when a single link failure occurs in the network and the network topology and traffic demands of the network are known in advance. The case of node failure (i.e. simultaneous failures of all links in and out of a node) and the case of capacity modularity are not treated in this thesis, thus are left as future work.

1.6 Organization of the Thesis

This thesis is organized as follows:

Chapter 2 discusses some basics of ATM networks, the importance of ATM networks, survivable networks and network restoration strategies. It also covers some basics of mathematical optimization as used in the context of this thesis. Moreover, previous research works related to our research problem are also discussed.

Chapter 3 describes our proposed mathematical models for the CFA problem and the proposed solution approaches for solving them.

Chapter 4 is devoted to the performance evaluation of our solution approaches. The global and failure-oriented reconfiguration strategies are compared based on predefined performance metrics.

Chapter 5 summarizes the overall contributions of our thesis while discussing possible future research directions.

Chapter 2

Background and Literature Review

In this chapter, some basic terminologies of our thesis are presented, which include: basic features of Asynchronous Transfer Mode (ATM) network, architecture of ATM network, routing in ATM network, ATM layers, self-healing network, network restorations, restoration schemes and mathematical optimization concepts. Some previous research works related to our research field are also discussed.

2.1 Background

2.1.1 Asynchronous Transfer Mode Protocol

ATM is a cell relay protocol, adopted by the ITU-T. ATM is a technology that provides a single platform for the transmission of voice, video and data at specified quality of service and speed. Some of the basic features of ATM are described as follows.

The basic unit of data in ATM network is a cell. A cell is only 53 bytes long with 5 bytes allocated to header and 48 bytes carry payload [7] as shown in Figure 2.1.

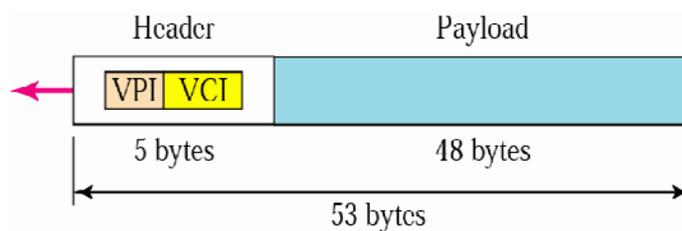


Figure 2.1: An ATM cell [7]

The header of the ATM cell is occupied by the virtual path identifier (VPI) and the virtual circuit identifier (VCI) that define the virtual connection through which a cell should travel from an endpoint to a switch or from a switch to another switch.

ATM uses asynchronous time division multiplexing as illustrated in Figure 2.2 to multiplex the cells coming from different channels. ATM uses fixed size slots and ATM multiplexers fill a slot with a cell from any input channel that has a cell; the slot is empty if none of the channels has a cell to send.

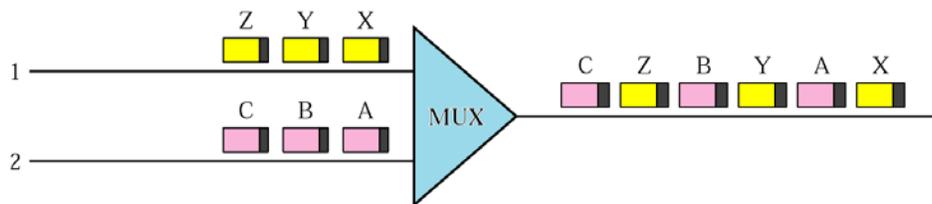


Figure 2.2: ATM Multiplexing [7]

ATM is a connection-oriented technology. Connection between two endpoints is accomplished through transmission paths (TPs), virtual paths (VPs), and virtual circuits (VCs). A transmission path (TP) is the physical connection (wire, cable, satellite, etc) between an endpoint and a switch or between two switches.

A transmission path is divided into several paths as shown in Figure 2.3. A virtual path (VP) provides a connection or a set of connections between two switches. ATM network is based on virtual circuits (VCs). All cells belonging to a single message follow the same virtual circuit and remain in their original order until they reach their destination.

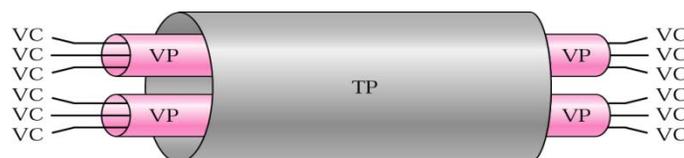


Figure 2.3: Relationship among the Transmission path, Virtual paths and Virtual circuits [7]

In ATM network, to route data from one endpoint to another, a virtual connection is defined by a pair of numbers: the virtual Path Identifier (VPI) and the virtual Circuit Identifier (VCI). The VPI defines the specific virtual path (VP), and the VCI defines a particular virtual circuit (VC) inside the VP. The VPI is the same for all virtual connections that are bundled into one VP as illustrated in Figure 2.4.

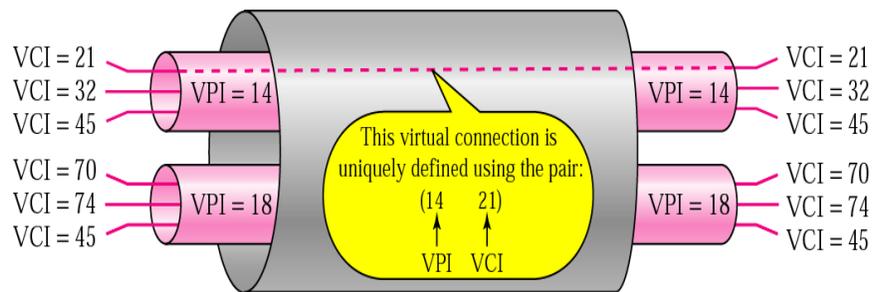


Figure 2.4: Connection identifiers in a Virtual Circuit of ATM network [7]

One of the unique capabilities of ATM technology is its ability to provide Quality of Service (QoS). The QoS of an ATM connection refers to limiting the cell loss, transit delay, and delay variation induced by the ATM network in transmitting the cells of that connection. In simple terms, guaranteeing the QoS is to ensure that a connection gets what it requires in terms of network resources.

2.1.1.1 ATM Network Architecture

ATM is a cell-switched network. The user access devices, called endpoints, are connected through a user-to-network interface (UNI) to the switches inside the network. The switches are connected through network-to-network interfaces (NNIs) as shown in Figure 2.5.

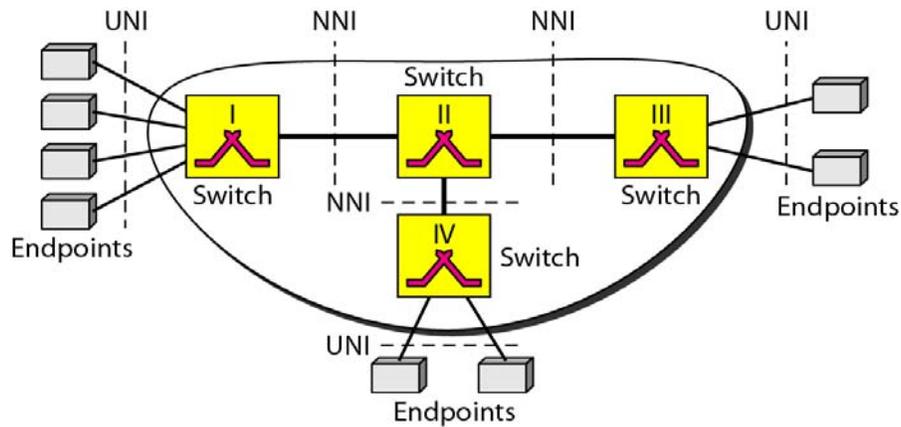


Figure 2.5: Architecture of an ATM network [7]

The ATM standard defines three layers: Application Adaptation Layer (AAL), ATM layer, and physical layer as shown in Figure 2.6.

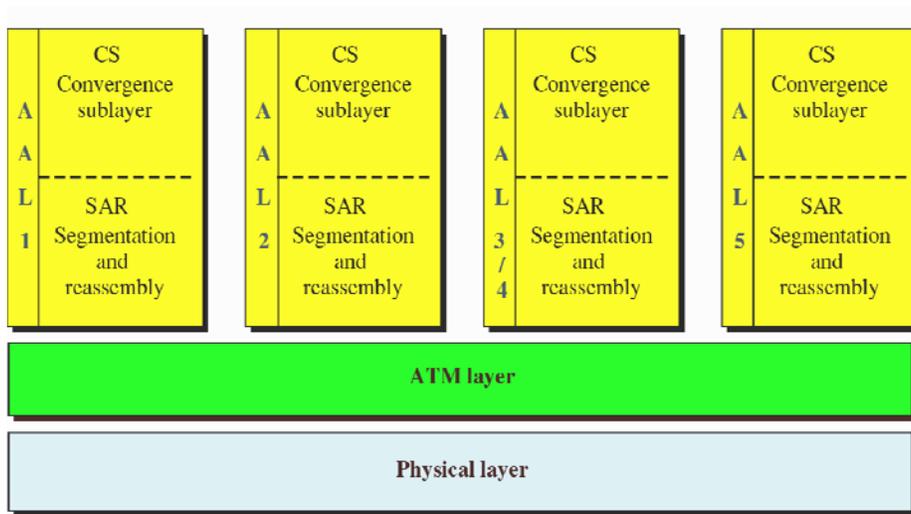


Figure 2.6: ATM layers [7]

- **Physical layer:**

ATM cells can be carried by any physical layer carrier. The original design of ATM is based on Synchronous Optical Network (SONET) as the physical layer carrier because of the high data rate of SONET and the clear definitions of the boundaries of cells in SONET. But

ATM does not limit the physical layer to SONET, other technologies, even wireless, may be used.

- **ATM layer**

The ATM layer provides routing, traffic management, switching, and multiplexing services. It process outgoing traffic by accepting 48-byte segments from the AAL sublayers and transforming them into 53-byte cells by adding of a 5-byte header.

- **Application Adaptation Layer (AAL)**

The AAL is designed basically to enable two ATM concepts which are: AAL segments the data from upper layer into 48-byte data units to be carried by a cell at the source and reassembles these segments at the destination to recreate the original payload by its segmentation and reassembly (SAR) sublayer. Before data are segmented by SAR, they must be prepared to guarantee the integrity of the data. This is done by a sublayer in AAL called the convergence sublayer (CS). ATM defines four versions of the AAL:

- AAL1, which supports applications that transfer information at constant bit rates such as video and voice.
- AAL2, which supports low bit rate traffic and short-frame traffic such as audio (example- mobile phone).
- AAL3/4, which supports connection-oriented and connectionless data. It provides comprehensive sequencing and error control mechanisms.
- AAL5, which supports applications that do not need comprehensive error control mechanisms.

2.1.1.2 Routing in ATM Network

ATM uses switches to route the cell from a source endpoint to the destination endpoint. A switch routes the cell using both the VPIs and the VCIs as illustrated in Figure 2.7.

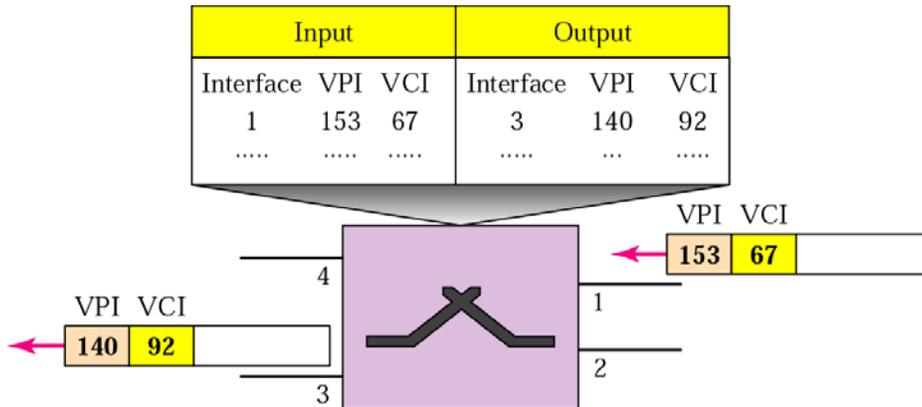


Figure 2.7: Routing in ATM network with a switch [7]

In Figure 2.7, a cell with a VPI of 153 and VCI 67 arrives at switch interface (port) 1. The switch checks its switching table, which stores six pieces of information per row: arrival interface number, incoming VPI, incoming VCI, corresponding outgoing interface number, the new VPI, and the new VCI. The switch finds the entry with the interface 1, VPI 153, and VCI 67 and discovers that the combination corresponds to output interface 3, VPI 140, and VCI 92. It changes the VPI and VCI in the header to 140 and 92, respectively, and sends the cell out through interface 3.

2.1.1.3 Current Transport Control Plane

The vision of designing a network with an integrated view of all the reliability mechanisms embedded was a big challenge from an implementation view point up to recent time. This situation has changed since the Internet Engineering Task Force (IETF) has introduced the Generalized Multiprotocol Level Switching (GMPLS) technology. In the current

overlay transport networks, IP/ATM/SONET, each layer manages its own control plane which can imitate restoration independently of what happens in other layers. With the advent of GMPLS, a new unified control plane and signalling functions is introduced, which is used for the design of all of the above transport networks including next generation networks. As such, the already installed ATM core resource management features must be reused as a particular implementation of GMPLS, either directly or with some adaptations. Among such transferable research works are studies related to the problem of capacity allocation and flow assignment in self-healing ATM networks [8]. In our thesis, we aim to address this problem by proposing and implementing two mathematical models.

2.1.2 Self-Healing Networks

Networks with fault tolerant capability are called self-healing networks. Self-healing or survivable network design refers to the incorporation of survivability strategies into the network design phase in order to mitigate the impact of a set of specific failure scenarios [4]. Capacity assignment is the major component in dimensioning a self-healing network when the network topology is given because sufficient redundant capacity must be preallocated in the network to quickly restore affected traffic via traffic rerouting upon network failure. In our thesis, we optimize the capacity and flow assignment for designing a self-healing ATM network while minimizing the network cost and fulfilling all traffic constraints.

2.1.3 Network Restoration

Traditional network survivability techniques have two aspects, survivable network design and network restoration [9]. These two phases are complementary to each other and cooperate to achieve seamless service upon failures. In network restoration, affected traffic demands are

rerouted upon failure to backup paths that have enough spare capacity provisioned in the self-healing network design phase.

Restoration schemes used in self-healing networks can in general be classified into two categories: preplanned restoration scheme and reactive restoration scheme.

In preplanned restoration scheme, all restoration routes with sufficient bandwidth are precomputed. In case of a failure, a node responsible for restoring the affected traffic simply activates the restoration routes and reroutes the affected traffic. Preplanned restoration can be used in the network environment where traffic patterns do not change frequently [3].

A reactive restoration scheme is a scheme that starts to search for alternate route with sufficient spare capacity after failure occurs by broadcasting restoration messages. So, in this case, the restoration routes are established immediately after the failure occurs in the network. Reactive restoration scheme is more suitable for networks with rapid change of traffic patterns. This restoration scheme requires little knowledge about current status of the network [3].

Compared to reactive restoration scheme where no spare capacity is preallocated before failure, pre-planning spare capacity not only guarantees service restoration, but also minimizes the duration and range of the failure impact. In packet-switched networks such as ATM, such service guarantees are especially important because backlog traffic accumulated during the failure restoration phase might introduce significant congestion [10], [11]. Preplanning spare capacity can mitigate or even avoid this congestion. On the other hand, reserving additional spare capacity increases the network redundancy in preplanned restoration scheme and preplanned restoration scheme is less flexible in handling unpredicted failure scenarios [4]. We should mention here that our proposed preplanned restoration strategies aim at providing a cost-effective capacity reservation at a 100% restoration level for a self-healing ATM network.

Depending on the location where traffic rerouting is performed, the restoration strategies in preplanned restoration scheme can be categorized into two classes: path restoration and link restoration strategies. More specifically, as our research scope is the path restoration strategy only, we have defined the path restoration strategy in detailed.

2.1.3.1 Link Restoration

The link restoration strategy is a local restoration scheme where the end nodes of the failed link make decision to reroute the affected traffic. By link failure, we mean all communication capacities between the two end nodes directly connected by the failed link are completely lost [3].

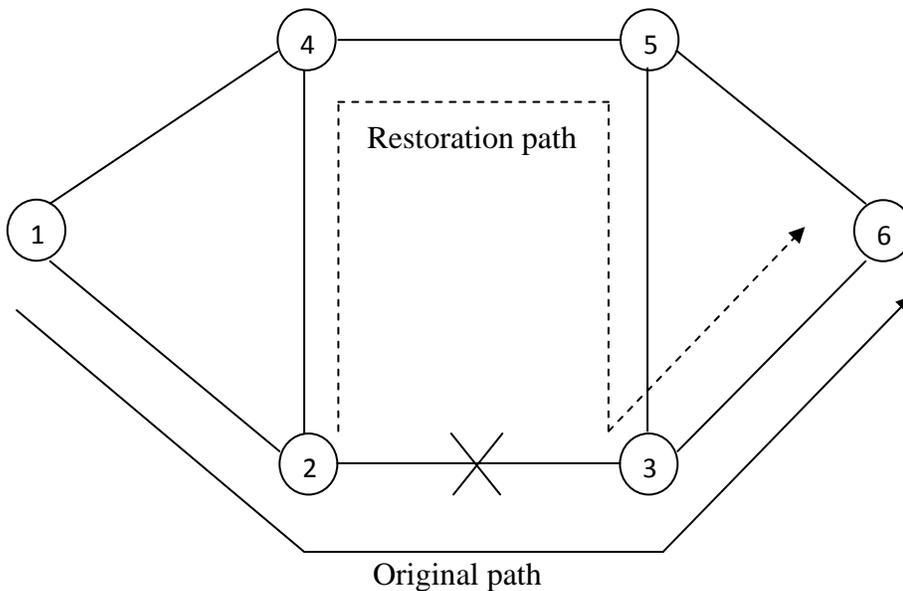


Figure 2.8: Link Restoration Strategy

In Figure 2.8, the link between the node 2 and node 3 has been affected for the failure. For the failure recovery, the end nodes of the failed link i.e. node 2 and node 3 have rerouted the affected traffic through 2-4-5-3.

2.1.3.2 Path Restoration

A path restoration strategy consists to setup backup paths between origin and destination nodes of the failed path when a failure occurs in the network. In this restoration scheme, the origin and destination nodes of the failed path make decision to reroute the affected traffic flow on the appropriate backup path [3]. The following Figure 2.9 depicts the path restoration strategy.

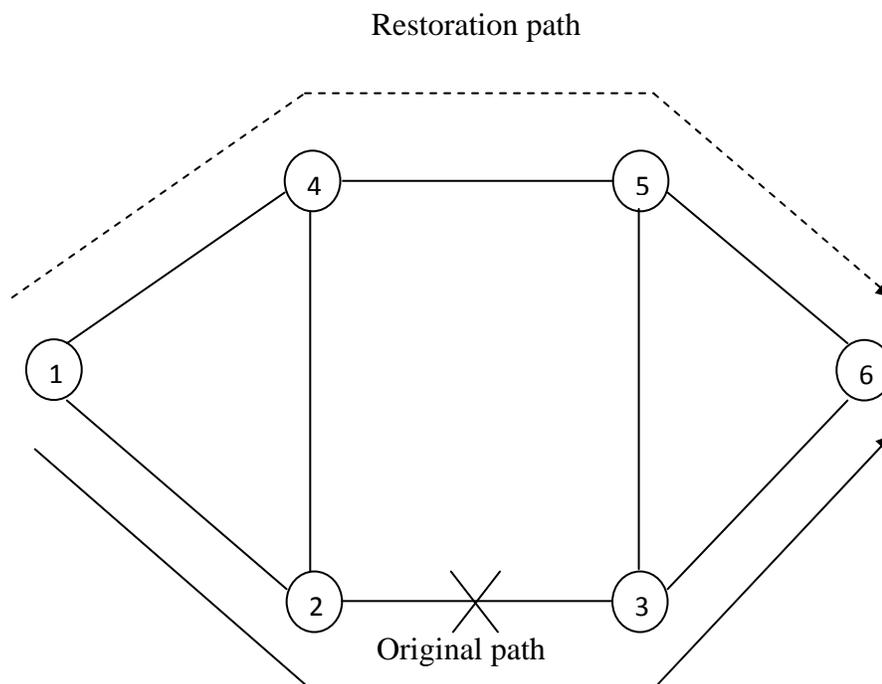


Figure 2.9: Path Restoration Strategy

In Figure 2.9, the path between the origin node 1 and destination node 6 i.e. 1-2-3-6 has been affected for the failure of the link between node 2 and node 3. For the failure recovery, the origin node 1 and destination node 6 have rerouted the affected traffic through the alternate backup path 1-4-5-6. This is an example of path restoration strategy where the backup path is constructed between the origin and destination nodes of the affected original path.

Path restoration strategy can further be classified into: Global reconfiguration and Failure-oriented reconfiguration strategies.

- In global reconfiguration strategy, in case of failure, all traffic flows, affected and unaffected, are reconfigured so that the restoration ratio is guaranteed but the total network cost is minimized [3]. In other words, in global reconfiguration strategy, the whole layout of the working paths (affected and unaffected) may be rearranged to overcome a failed link or node.
- In the failure-oriented reconfiguration strategy, only the affected traffic flows are rerouted at failure events. Unaffected traffic flows remain unchanged.

The categories of the restoration scheme can be summarized as shown in Figure 2.10.

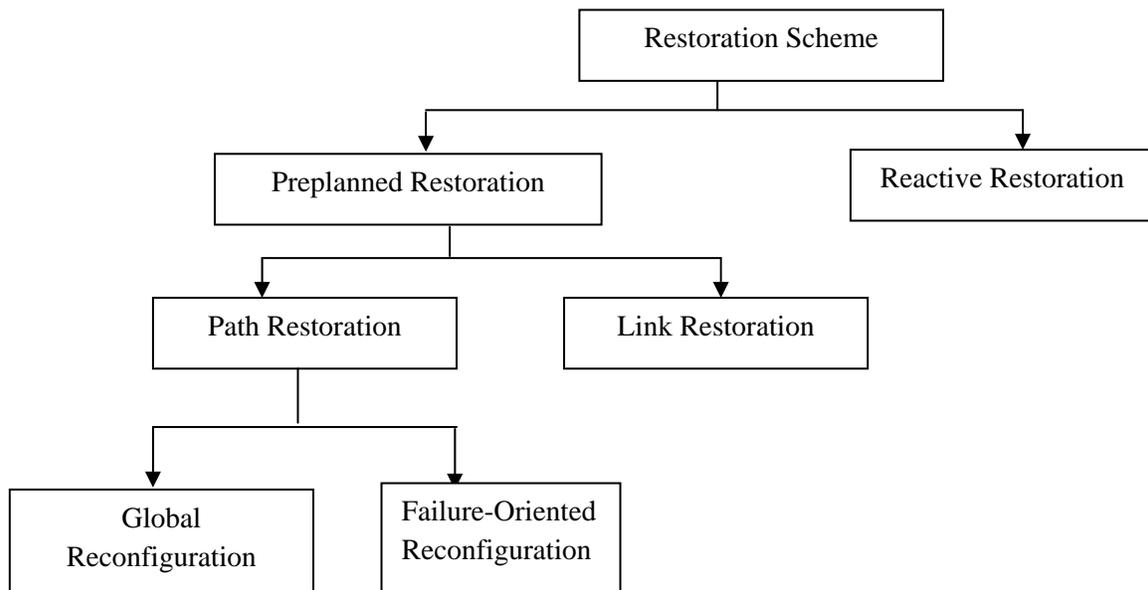


Figure 2.10: Categories of Restoration Schemes for a self-healing network

We should mention here that two nonlinear mathematical models for the global and failure-oriented reconfiguration strategies have been proposed in this thesis for designing a self-healing ATM network. The details of our proposed strategies are discussed in chapter 3.

2.1.4 Mathematical Optimization Techniques

Mathematical optimization (also called mathematical programming) is a field in applied mathematics. The aim of mathematical optimization is to find a minimum or a maximum of a given function. Optimization plays an important role in many branches of science and applications such as network analysis, economics, optimal design of mechanical systems, etc.

Mathematical models of optimization can be generally represented by a set of constraints X and a cost function f that maps the elements of X into real numbers. The set X consists of decision variables x . The general form of an optimization problem can be written as follows:

$$(P): \min f(x) \tag{2.1}$$

s.t.

$$g(x) = 0$$

where $f(x)$ is called the objective function, $g(x)$ defines the constraints of the problem (P) and x is a vector in \mathbb{R}^n . In fact, we want to find an optimal decision variable i.e., $x^* \in X$ such that $f(x^*) \leq f(x), \forall x \in X$.

2.1.4.1 Linear Optimization Problems

When the objective function in (2.1) and the constraint are linear, the optimization problem is called linear optimization problem. If the variables of the objective function and the constraints are integers, the problem is called linear Integer problem. The optimization problem

is called mixed linear integer problem when some of the variables of the objective function are integers and the others are continuous.

2.1.4.2 Nonlinear Optimization Problems

When the objective function and/or the constraint of problem (P) in (2.1) are nonlinear, the optimization problem is called a nonlinear optimization problem. In the nonlinear optimization problem, the variables of the objective function can also be integer and in that case the problem is called a nonlinear integer optimization problem. The optimization problem is called a mixed integer nonlinear optimization problem, when the variables can be both integer and continuous.

A variable x which verifies the constraints (i.e. such that $g(x) = 0$ in (2.1)) is called a feasible solution. A feasible set is the set of all feasible solutions for an optimization problem.

2.1.4.3 Unconstrained Nonlinear Optimization Problems

An optimization problem such as problem (P) in (2.1), but without the mathematical constraints is called unconstrained optimization problem. There are several efficient methods for solving unconstrained nonlinear optimization problems.

Suppose we want to solve the following unconstrained optimization problem where the problem is named as problem (P_1) such as: $\min f(x)$ where $x \in \mathbb{R}$ and $f(x)$ is twice differentiable. The numerical iterative methods for solving the problem (P_1) aim to find the optimal value x^* and are known as gradient methods. These methods basically work as follows:

Steps of solving Unconstrained Optimization Problem

Step 1: All the necessary parameters are initialized as follows:

$\varepsilon > 0$, x_0 and $t = 0$. Here, ε is chosen as a very small value, x_0 is the initial value of the decision variable.

Step 2: If $\|\nabla f(x_t)\| < \varepsilon$ and $\nabla^2 f(x_t)$ is positive definite, then stop. Here, $\|x_t\|$ is the vector norm of x . Else go to Step 3.

Step 3: Calculate $x_{t+1} = x_t + \alpha_t d_t$ and $t = t + 1$ and go back to Step 2.

In this step, d_t is a descent direction because it decreases the value of the objective function. Several descent directions are proposed and each of them corresponds to specific method with different convergence rate (so-called speed of convergence). When $d_t = -\nabla f(x_t)$, the method is called steepest descent method. When $d_t = -[\nabla^2 f(x_t)]^{-1} \nabla f(x_t)$, the gradient method is called Newton method. Sometimes, $\nabla^2 f(x_t)$ is not positive defined or does not exist. In that case, d_t is not a descent direction and $\nabla^2 f(x_t)$ is modified with a definite positive matrix. This involves a modified Newton method.

α_t in the step 3, is called the step size in the descent direction and several methods are used to find a suitable step size in the descent direction such as the Armijo method. The convergence analysis [12] shows that the rate of convergence is not the same for all gradient methods. The Newton method is the fastest among all methods [12] and several modifications of this method are proposed in the literature for the speed of convergence. But the drawback of the Newton method is its local convergence. In this thesis, we use both the modified Newton method and the Armijo method.

2.1.4.4 Constrained Nonlinear Optimization Problems

In a constrained optimization problem, both the objective function and the constraint are differentiable. The resolution of the optimization problem becomes much more difficult because of the presence of constraints. For solving constrained nonlinear optimization problems, usually Penalty methods, Lagrangian methods or Augmented Lagrangian methods are used where the constrained optimization problems are converted to unconstrained problems. As our proposed restoration strategies deal with a constrained nonlinear optimization problem, a description of the basics of Penalty methods, Lagrangian methods and Augmented Lagrangian methods follows. These methods in fact convert the constrained optimization problem into an unconstrained optimization problem and then solve it indirectly that way.

2.1.4.5 Penalty Methods

The basic idea of penalty methods is to eliminate the constraints of the optimization problem by adding them with the objective function via the use of penalty parameters that penalize any violation of the constraints. In penalty methods, the penalty parameters determine the severity of the penalty and as a consequence, the extent to which the resulting unconstrained problem approximates the original constrained problem. As penalty parameters take higher values, the approximation becomes increasingly accurate [12].

There are several penalty methods proposed in the literature. Only one of the penalty methods called exact penalty method is defined in the following.

In the exact penalty method, the optimization problem (P) in (2.1) is transformed to an equivalent problem as follows:

$$\min f(x_t) + \frac{\lambda_t}{2} \|g(x_t)\|^2 \text{ where } x \in \mathbb{R} \quad (2.2)$$

In (2.2), λ_t is a penalty parameter. The sequence of λ_t converges to infinity when $t \rightarrow \infty$.

When $\lambda_t \rightarrow \infty$, this situation enforces the solution of the optimization problem to be feasible.

For solving the problem described in (2.2), any unconstrained optimization method can be used. Typically, the solution steps of the unconstrained problem are as follows:

Penalty Methods

Step 1: The parameters are chosen as $\lambda_0 > 0$ and $t = 0$.

Step 2: The Equation (2.2) i.e. $\min f(x_t) + \frac{\lambda_t}{2} \|g(x_t)\|^2$ is solved with an unconstrained method.

Then Step 3 is executed.

Step 3: If the solution achieved from Step 2 is optimal, then the process stops. Otherwise it goes to Step 4.

Step 4: The penalty parameter λ_t is updated and the process is repeated starting from Step 2.

The penalty parameter λ_t is chosen in a way so that it converges to infinity, when $t \rightarrow \infty$

Updating of the penalty parameter can be done in several ways, such as $\lambda_{t+1} = \beta\lambda_t$ where $\beta > 1$.

2.1.4.6 Lagrangian Method

In Lagrangian method [12], the nonlinear constrained optimization problem (P) described in (2.1) is replaced by the following unconstrained optimization problem as follows:

$$\min f(x_t) + u_t g(x_t) \text{ where } x \in \mathbb{R} \quad (2.3)$$

In (2.3), u_t is a sequence called Lagrange multipliers or dual variables and $f(x_t) + u_t g(x_t)$ is called the Lagrangian function.

Typically, the Lagrangian method works as follows:

Lagrangian Algorithm

Step 1: Initialize u_0 and set $t = 0$.

Step 2: The unconstrained problem $\min f(x_t) + u_t g(x_t)$ is solved using an unconstrained method.

Step 3: If the solution achieved from Step 2 is optimal, then the method is stopped. Otherwise, go to Step 4.

Step 4: The Lagrange multiplier u_t is updated and the method is repeated starting from Step 2.

2.1.4.7 Augmented Lagrangian Methods

Augmented Lagrangian methods [12] are also referred to as multipliers methods. Augmented Lagrangian methods are hybrid methods between Lagrangian method and Penalty methods. Here, we discuss the Augmented Lagrangian method which uses the exact penalty term. In this method, the constrained optimization problem (P) in (2.1) is transformed as

$$\min f(x_t) + u_t g(x_t) + \frac{\lambda_t}{2} \|g(x_t)\|^2 \tag{2.4}$$

Here, u_t is the Lagrange multiplier and λ_t is the penalty parameter. But, the penalty parameter λ_t does not need to go to infinity. The steps of the Augmented Lagrangian methods can be described as follows:

Augmented Lagrangian Algorithm

Step 1: $\lambda_0 > 0$, and $t = 0$ are initialized and u_0 is chosen.

Step 2: The problem $\min f(x_t) + u_t(g(x_t)) \frac{\lambda_t}{2} \|g(x_t)\|^2$ is solved with an unconstrained solution method.

Step 3: If the solution from Step 2 is optimal, the method is stopped. Otherwise go to Step 4.

Step 4: u_t and λ_t are updated as shown below and the method goes back to Step 2.

- $\lambda_{t+1} = \beta \lambda_t$
- $u_{t+1} = u_t + \lambda_t g(x_t)$
- $t = t + 1$

We should mention that, the above stop criterion of the Augmented Lagrangian method is based on the Karush-Khun-Tucker optimality conditions (KKT conditions) [12].

2.2 Literature Review

Fast restoration from a network failure has been recognized as one of the key issues for high-speed networks such as ATM [13]. A lot of attention has been paid to ATM technology because of its features. Many research works have been dedicated to the design of self-healing ATM networks.

The problem of designing self-healing ATM networks is mostly formulated as a linear multicommodity flow problem in the literature where the objective function is to minimize the spare capacity cost which is used for network traffic restoration when a network fails. Only few research works have treated the case where the objective function is nonlinear.

In [3], Xiong and Mason investigated the CFA problem in the design of self-healing ATM networks as a linear programming problem using the Virtual path (VP) concept. In their work, preplanned restoration schemes in the design of self-healing ATM networks under a single link or node failure scenario, with restoration performed on the VP level are considered. Minimizing the spare capacity cost for a given restoration requirement is their main objective. Assuming the spare capacity as the main cost, the CFA problem is modeled as the optimization of capacity allocation and flow assignment in order to minimize the spare capacity cost while meeting the network survivability requirements. Several restoration strategies are compared quantitatively in terms of spare cost, namely: global versus failure-oriented reconfiguration, path versus link restoration, and state dependent versus state independent restoration. The advantages and disadvantages of the various restoration strategies are highlighted in this paper. However, no capacity modularity to conform to physical requirements was introduced in their model.

In [14], Woungang et al. also investigated the Capacity and Flow assignment- Network Survivability (CFA-NS) problem as a linear programming problem. They quantitatively compare the network survivability for link and path restorations, under various traffic and design patterns and spare capacity distribution schemes using the aggregate restoration ratio as performance metric. Even though their work provides a solid foundation for the design of restorable ATM mesh-type networks, the case of node failure was not treated.

In [15], Lee and Koh proposed the design of self-healing ATM networks based on back up VPs using an integer linear programming model. Assuming that the bandwidth requirements and original routes for the target traffic demands are given, the main goal of this paper is to find the backup routes of each traffic demand that is link disjoint from the working route. The authors argued that when using their model, the worst case survivability could be improved significantly

with relatively small increase of bandwidth. However, the method based on which this improvement could be achieved was not disclosed.

In [1], Kwong et al. studied the CFA problem arising in the design of self-healing ATM networks using the Virtual path (VP) concept. The authors introduced a genetic algorithm (GA) to implement the backup path search process instead of using the trivial exhaustive search method. Indeed, their method finds the set of multicast backup virtual paths (BVPs) to recover the traffic flows of the multicast primary virtual paths (PVPs) that are affected when a link failure occurs in the network. The major drawback of this approach is that there is no guarantee that their method can generate a global optimum in real time.

In [16], Wu et al. proposed a Virtual Path (VP) based preplanned restoration scheme for self-healing ATM networks, with the target of minimizing the spare capacity usage. An analysis of several backup path building schemes for multicast tree in self-healing ATM networks is also provided and a self-healing scheme which is a modified line protection scheme for multicast communication failure is introduced.

In [4], Liu et al. proposed an integer programming model for the CFA problem. In their model, the Spare Capacity Allocation (SCA) structure is modeled as a matrix method and an adaptive algorithm is developed to approximate the optimal SCA solution termed as Successive Survivable Routing (SSR) algorithm. The authors pointed out that, their algorithm could provide a near optimal spare capacity allocation, not the optimal spare capacity allocation.

In [17], Kawamura et al. proposed a high-speed self-healing scheme to improve the reliability of virtual path based ATM networks. The characteristics of virtual paths and their influence on failure restoration are also discussed. The proposed self-healing scheme has several

advantages such as restoration rapidity, reduction in the amount of spare resources required, node failure restoration, simplified message transmission processes, reduced number of generated messages by using preassigned backup virtual paths. The authors mentioned that a more effective reconstruction cycle should be realized with an efficient spare resource designing algorithm, but this algorithm was not developed in their paper.

In [5], Murakami and Kim also investigated the CFA problem for fully restorable ATM networks as a large scale linear programming problem. A joint optimization method for the survivable flow and capacity assignment problem was presented given the projected traffic demand. However, due to the mutual dependency between the capacity placement and the survivable flow assignment, their proposed solution cannot not be claimed to be optimum when the CFA problem is treated separately.

In [2], Murakami and Kim provided an extensive comparative analysis of end-to-end and line restoration schemes based on the required spare capacity cost. The benefits of end-to-end restoration scheme quantitatively in terms of minimum resource installation cost are also discussed, and a joint capacity and flow optimization algorithm is proposed that can achieve fully restorable ATM networks based on end-to-end and link restoration using a linear programming model. The major drawback of this paper is, a joint capacity and flow optimization method is developed for the fully restorable networks based on the end-to-end restoration, so this method cannot be claimed optimum if the capacity and flow are optimized separately.

In [18], Murakami and Kim also investigated the CFA problem and developed joint optimization methods of capacity and flow assignment for solving this problem. Link restoration and end-to end restoration are compared with respect to the required spare capacity cost, revealing that the regularity of a network is a critical factor when determining the redundant

capacity cost. But the proposed optimization method in this paper is a joint method, which does not treat the optimal capacity and flow assignment separately.

In [19], Murakami and Kim investigated the problem of optimal Virtual Path routing for survivable ATM networks, called as a Survivable virtual path routing (SVPR) problem. In their paper, an algorithm is proposed to find a virtual path configuration and bandwidth assignment that minimizes the expected amount of lost flow upon restoration of a network failure. The SVPR problem is formulated as a nonlinear, non smooth multicommodity flow problem with linear constraints. A modified flow deviation method is developed to obtain a near-optimal solution of the problem. The two major drawbacks of this proposed method are: the optimization parameters have to be adjusted properly specially for the heavily loaded situation of the network to prevent premature convergence and the proposed method can achieve better survivability only for a sparse network.

The above discussed papers [3-5] and [14-16], addressed the CFA problem as a linear multicommodity flow problem for survivable ATM networks. To the best of our knowledge, only few papers have treated the same problem as a nonlinear multicommodity flow problem (including the work in [17]), but with the goal of minimizing the spare capacity cost only, assuming that the working capacity is given. Unlike previous works, our proposed models consider the aggregation of routing cost and capacity installation cost in the objective functions of our nonlinear models. The objective is to minimize this aggregated cost while satisfying all traffic constraints.

Chapter 3

Optimization Models and Solution Approaches

In this chapter, we formulate our research problems. We also describe our proposed mathematical models for the global reconfiguration and the failure-oriented reconfiguration strategies. Finally, the methods developed for solving the proposed mathematical models are presented.

3.1 Mathematical Model for the Global Reconfiguration

Strategy

In global reconfiguration strategy, the affected and unaffected traffic flows are both reconfigured when a failure occurs in the network. To describe the proposed model, we use the notations captured in Table 3.1.

Table 3.1: Notations used in the mathematical model for the Global Reconfiguration Strategy

Notations	Meaning
$N(A, B)$	Network is represented by a graph N , where A is the number of nodes and B is the number of arcs
$(n = A)$	Total number of arcs
$a = 1, 2, 3, \dots, n$	Each arc of the network is denoted by a
d_a	Unit cost of capacity installation
γ_a	Unit cost of transit delay
K	Number of Commodities (source-destination pairs)
N_k	Number of paths for commodity k
r_k	Traffic demand (bandwidth requirement) of commodity k
$S - s_0$	Set of Failure states of the network (s_0 is the non failure state)

x_a^s	Aggregate flow on the arc a when the network is in state s
C_a^s	Capacity on arc a when the network is in state s
π_{kp}^a	= 1, if the p^{th} path of commodity K uses the arc a and 0 otherwise
x_{kp}^s	The amount of flow commodity k using by its p^{th} path when the network is in the state $s \in S$

Our proposed mathematical model for the capacity and flow assignment problem (CFA) with a global reconfiguration strategy can be formulated as following:

$$\min \sum_{a=1}^n [\gamma_a \frac{x_a^s}{C_a^s - x_a^s} + d_a C_a^s] \quad (3.1)$$

s.t.

$$\sum_{k=1}^K \sum_{p=1}^{N_k} \pi_{kp}^a x_{kp}^s = x_a^s, \forall a = 1, \dots, n, \forall s \in S \quad (3.2)$$

$$\sum_{p=1}^{N_k} x_{kp}^s = r_k, \forall k = 1, \dots, K, \forall s \in S \quad (3.3)$$

$$0 \leq x_a^s \leq C_a^s, \forall a = 1, \dots, n, \forall s \in S \quad (3.4)$$

$$x_{kp}^s \geq 0, \forall K = 1, \dots, K, \forall p = 1, \dots, N_k \quad (3.5)$$

Equation (3.1) is the objective function of the proposed global reconfiguration strategy. The first term of the objective function represents the routing cost given by Kleinrock's average delay function [20] and the second term is the cost of the installed capacities.

- Constraint (3.2) in fact denotes the aggregated flow constraints. It ensures that the aggregated flow on each arc a should be equal to the sum of flows using that arc by all the paths of all commodities.

- Constraint (3.3) guarantees that the amount of flow of a specific commodity using all its paths should be equal to the required bandwidth requirement of that commodity.
- Constraint (3.4) ensures that the total flow on each arc a should be smaller than the capacity of that arc.
- Constraint (3.5) ensures that the traffic flow of every commodity on all of its paths should be positive.

3.2 Mathematical Model for the Failure-Oriented Reconfiguration Strategy

In failure-oriented reconfiguration strategy, only the affected traffic flows are rerouted in case a link fails in the network. The origin and destination nodes of the affected paths make decision about the rerouting of the affected flows for failure recovery.

To describe the mathematical model in this case, we use the notations shown in Table 3.2.

Table 3.2: Notations used in the mathematical model for the Failure-Oriented Reconfiguration Strategy

Notations	Meaning
$N(A, B)$	Network is represented by a graph N , where A is the number of nodes and B is the number of arcs
$(n = A)$	Number of arcs
$a = 1, 2, 3, \dots, n$	Each arc of the network is denoted by a
d_a	Unit cost of capacity installation
γ_a	Unit cost of Transit delay
K	Number of Commodities (source-destination pairs)
M_k	Number of backup paths for commodity k
r_k	Traffic demand (bandwidth requirement) of commodity k
$S - s_0$	Set of Failure states of the network (s_0 is the non failure state)

$x_a^{s_0}$	Aggregated non affected traffic flow on the arc a when the network is in state s
π_{kp}^a	Equals to 1 if the p^{th} path of commodity k uses the arc a and 0 otherwise
C_a^s	Capacity on arc a when the network is in state s
$x_{kp}^{s_0}$	The amount of unaffected flow commodity k using by its p^{th} path when the network is in the state $s \in S$
y_a^s	Aggregated flow rerouted on arc a
μ_{kp}^s	= 1, if the p^{th} working path of the commodity k is affected when the network is on state s , otherwise 0
α_{kq}^a	= 1, if the q^{th} backup path of commodity k uses the arc a , otherwise 0
y_{kq}^s	The amount of restored flow commodity k passing through its back up path q when the network is in state s
z_a^s	Total aggregated flow on the arc a i.e. $z_a^s = x_a^{s_0} + y_a^s$

The capacity and flow assignment (CFA) problem with a failure-oriented reconfiguration strategy can be formulated as follows:

$$\min \sum_{a=1}^n \left[\gamma_a \frac{x_a^{s_0} + y_a^s}{C_a^s - (x_a^{s_0} + y_a^s)} + d_a C_a^s \right] \quad (3.6)$$

s.t.

$$\sum_{k=1}^K \sum_{p=1}^{N_k} \pi_{kp}^a (1 - \mu_{kp}^s) x_{kp}^{s_0} = x_a^{s_0}, \forall a = 1, \dots, n \quad (3.7)$$

$$\sum_{k=1}^K \sum_{p=1}^{N_k} \alpha_{kq}^a y_{kq}^s = y_a^s, \forall a = 1, \dots, n, \forall s \in S \setminus \{s_0\} \quad (3.8)$$

$$\sum_{p=1}^{N_k} x_{kp}^{s_0} = r_k, \forall k = 1, \dots, K \quad (3.9)$$

$$\sum_{q=1}^{M_k} y_{kq}^s = \sum_{p=1}^{N_k} \mu_{kp}^s x_{kp}^{s_0}, \forall k = 1, \dots, K \quad (3.10)$$

$$0 \leq x_a^{s_o} + y_a^s < C_a^s, \forall a = 1, \dots, n, \forall s \in S \quad (3.11)$$

$$x_{kp}^{s_o} \geq 0, \forall k = 1, \dots, n, \forall p = 1, \dots, N_k \quad (3.12)$$

$$y_{kq}^s \geq 0, \forall k = 1, \dots, n, \forall q = 1, \dots, M_k \quad (3.13)$$

Equation (3.6) is the objective function where the first term represents the average delay function (routing cost) [20] and the second term is the capacity installation cost.

- Constraint (3.7) ensures that the aggregated unaffected traffic flow of an a is the summation of the unaffected flows of all the paths of all the commodities through the arc a .
- Constraint (3.8) depicts the aggregation of the amount of affected traffic flow that is rerouted on arc a .
- Constraint (3.9) is to ensure that all traffic demands for all the commodities are met.
- Constraint (3.10) ensures that all the affected flow of the working routes of the network are rerouted through the backup paths.
- Constraint (3.11) assures that enough bandwidth should be reserved in the network to reroute the traffic flow affected by a failure.
- Constraints (3.12) and (3.13) are positivity constraints which ensure that the unaffected traffic flows and the reroute traffic flows should be positive.

By setting $z_a^s = x_a^{s_o} + y_a^s$, the above model represented by the Equations (3.6) to (3.13) can be reformulated as:

$$\min \sum_{a=1}^n \left[\gamma_a \frac{z_a^s}{C_a^s - (x_a^{s_o} + y_a^s)} + d_a C_a^s \right] \quad (3.14)$$

s.t.

$$z_a^s = \sum_{k=1}^K \left[\sum_{p=1}^{N_k} \pi_{kp}^a (1 - \mu_{kp}^s) x_{kp}^{s_o} + \sum_{q=1}^{M_k} \alpha_{kq}^a y_{kq}^s \right], \forall a = 1, \dots, n, \forall s \in S \quad (3.15)$$

$$\sum_{p=1}^{N_k} x_{kp}^{s_o} = r_k, \forall k = 1, \dots, K \quad (3.16)$$

$$\sum_{q=1}^{M_k} y_{kq}^s = \sum_{p=1}^{N_k} \mu_{kp}^s x_{kp}^{s_o}, \forall k = 1, \dots, K \quad (3.17)$$

$$0 \leq x_a^{s_o} + y_a^s < C_a^s, \forall a = 1, \dots, n, \forall s \in S \quad (3.18)$$

$$x_{kp}^{s_o} \geq 0, \forall k = 1, \dots, n, \forall p = 1, \dots, N_k \quad (3.19)$$

$$y_{kq}^s \geq 0, \forall k = 1, \dots, n, \forall q = 1, \dots, M_k \quad (3.20)$$

We now describe the methods that we use to solve the models represented by the Equations (3.1) to (3.5) and (3.14) to (3.20).

3.3 Solution Approaches

A special case of the Augmented Lagrangian Algorithm (so-called Separable Augmented Lagrangian Algorithm (SALA)) is proposed to solve the aforementioned models. We have used the Modified Newton method to calculate the optimal flows and capacities of the arc in our SALA based restoration strategies. The Armijo method is used to determine the step size of the descent direction of the Modified Newton method and the Dijkstra algorithm is used for solving the shortest path problems.

3.3.1 Separable Augmented Lagrangian Algorithm (SALA)

The Separable Augmented Lagrangian Algorithm (SALA) was introduced by Hamdi [21] for solving separable nonlinear large-scale programming problems. For a better understanding of the SALA, let's consider the following separable problem:

$$\min \sum_{j=1}^n f_j(x_j) \quad (3.21)$$

s.t.

$$\sum_{j=1}^n g_j(x_j) = 0 \quad (3.22)$$

$$x_j \in S_j, \quad \forall j = 1, \dots, n \quad (3.23)$$

where f_j functions are twice differentiable, x_j are n_j -vectors and S_j are closed and bounded subsets of \mathbb{R}^{n_j} . $g_j: \mathbb{R}^{n_j} \rightarrow \mathbb{R}^m$ are smooth functions and constraints (3.22) are coupling constraints. To exploit the separability of the objective function, SALA consists of introducing allocation vectors y_j to decouple the constraints (3.22) by setting $y_j = -g_j(x_j), \forall j = 1, \dots, n$. We obtain the following equivalent problem:

$$\min \sum_{j=1}^n f_j(x_j) \quad (3.24)$$

s.t.

$$\sum_{j=1}^n y_j = 0 \quad (3.25)$$

$$g_j(x_j) + y_j = 0, \quad \forall j = 1, \dots, n \quad (3.26)$$

$$x_j \in S_j, \quad \forall j = 1, \dots, n \quad (3.27)$$

Applying the Augmented Lagrangian method [12] on the above equivalent problem, we obtain the following equations:

$$\min \sum_{j=1}^n \left[f_j(x_j) + u_j(g_j(x_j) + y_j) + \frac{\lambda}{2} \|g_j(x_j) + y_j\|^2 \right] \quad (3.28)$$

s.t.

$$\sum_{j=1}^n y_j = 0 \quad (3.29)$$

$$x_j \in S_j, \forall j = 1, \dots, n \quad (3.30)$$

In (3.28), u_j are Lagrange multipliers and $\lambda > 0$ is a penalty parameter. Hamdi et al.[21] proposed to solve the problem in (3.28) with respect to x_j and y_j alternatively. It should be noted that, at iteration t , when the problem is solved with respect to one variable, the other one is assumed to be known. At iteration t , we then obtain the following sub problems:

$$x_j^t = \arg \min_{x_j \in S_j} \left\{ f_j(x_j) + u_j^t(g_j(x_j) + y_j^{t-1}) + \frac{\lambda_t}{2} \|g_j(x_j) + y_j^{t-1}\|^2 \right\} \quad (3.31)$$

$$y_j^t = \arg \min_{\sum_j y_j = 0} \left\{ u_j^t(g_j(x_j^t) + y_j) + \frac{\lambda_t}{2} \|g_j(x_j^t) + y_j\|^2 \right\} \quad (3.32)$$

$$u_j^{t+1} = u_j^t + \lambda_t (g_j(x_j^t) + y_j^t) \quad (3.33)$$

The Equation (3.33) is the Lagrange multiplier's update formula [12]. The sub problems in (3.32) can be solved explicitly by using the Karush-Kuhn-Tucker conditions. We then obtain,

$$y_j^t = -g_j(x_j^t) + \frac{1}{n} \sum_{j=1}^n g_j(x_j^t), \quad \forall j = 1, \dots, n \quad \text{and} \quad u^{t+1} = u^t + \frac{\lambda_t}{n} \sum_{j=1}^n g_j(x_j^t) \quad (3.34)$$

From (3.34) it can be observed that the same dual vector is used for all sub problems.

Now, we can describe the Separable Augmented Lagrangian Algorithm as follows:

Separable Augmented Lagrangian Algorithm (SALA)

1. $u^1, \lambda > 0, \varepsilon > 0, \beta > 0, y_j^0$ are chosen such that $\sum y_j^0 = 0$ and $t = 1$.

2. $\forall j = 1, \dots, n$ Compute

$$x_j^t = \arg \min_{x_j \in S_j} \left\{ f_j(x_j) + u_j^t (g_j(x_j) + y_j^{t-1}) + \frac{\lambda_t}{2} \|g_j(x_j) + y_j^{t-1}\|^2 \right\}$$

3. Compute $\delta^t = \sum_{j=1}^n g_j(x_j^t)$

If $\|\delta^t\| < \varepsilon$, the algorithm stops. Otherwise it goes to next step.

4. Update u, λ, y_j according to the following formulas:

$$y_j^t = -g_j(x_j^t) + \frac{1}{n} \delta^t$$

$$u^{t+1} = u^t + \frac{\lambda_t}{n} \delta^t$$

$$\lambda^{t+1} = \beta \lambda^t$$

$t \leftarrow t + 1$ and go back to step 2.

3.3.2 Dijkstra Algorithm

Dijkstra algorithm is a graph search algorithm that solves the single source shortest path problem for a graph with nonnegative path costs, producing a shortest path tree. This algorithm is often used in routing and as a subroutine in other graph algorithms.

For a given source node in a graph, the algorithm finds the path with the lowest cost (i.e. the shortest path) between the source node and every other node of the graph. It can also be used

for finding the costs of shortest paths from a source node to a destination node by stopping the algorithm once the shortest path to the destination node has been determined.

Our SALA-based algorithms use the Dijkstra algorithm to find the shortest paths for all the commodities (source-destination pairs). At each iteration, we try to explore a new shortest path for each commodity which is not explored in the previous iterations and whose cost is less than those of the previously explored shortest paths until we reach a satisfactory solution. The complexity of the Dijkstra algorithm is $O(|N|^2)$ where N is the number of nodes in the network.

3.3.3 Modified Newton Method

In our SALA based mathematical models, the modified Newton method is used as a descent method for solving the optimization problem. Its idea is to take advantage of the global convergence properties of the steepest descent method [22]. On the other hand, its quadratic convergence should be exploited when the iteration gets close enough to the optimal solution.

In our SALA based restoration strategies, the modified Newton method is used to solve the following optimization problem:

$$\min f(x_a, C_a) = \left\{ \frac{\gamma_a x_a}{C_a - x_a} + d_a C_a + u_a^t x_a + \frac{\lambda_t}{2} \left[(x_a)^2 - 2x_a \left(x_a^{t-1} - \frac{\delta_a^{t-1}}{n_a^t} \right) \right] \right\} \text{ where } 0 \leq x_a < C_a.$$

The associated Lagrangian function is given by $L(x_a, C_a, \mu_a) = f(x_a, C_a) + \mu_a(x_a - C_a)$ where $\mu_a \geq 0$ is the Lagrange multiplier associated to the unique constraint and x_a^{t-1} is the solution at iteration $t-1$. The modified Newton method is used for the minimization of the Lagrangian function.

The steps of the modified-Newton method are as follows:

Modified Newton Algorithm

Step 1: All the necessary parameters are initialized as follows:

$\varepsilon > 0, \mu_a^0, 0 \leq x_a^0 < C_a^0, \alpha \in [0;1], i = 0$ where, μ_a^0 is the initial value of the Lagrange multiplier, x_a^0 and C_a^0 are the initial flow and capacity of arc a , α is the step size of the descent direction, and i is the iteration number.

Step 2: The step size of the modified Newton method in the descent direction is calculated by the Armijo method.

$\alpha_i = \text{Armijo}(x_a^i, C_a^i, \mu_a^i, d_M^i)$ where, d_M is the descent direction of the modified Newton method.

Step 3: Calculate the capacity and flow of an arc for the next iteration using:

$$(C_a^{i+1}, x_a^{i+1}) = (C_a^i, x_a^i) + \alpha_i d_M^i$$

Step 4: If $\|\nabla L(x_a^{i+1}, C_a^{i+1}, \mu_a^i)\| < \varepsilon$ Stop (if the obtained capacity and flow are optimal)

Else set $\mu_a^{i+1} = \mu_a^i + \alpha(x_a^{i+1} - C_a^{i+1}), i = i + 1$, and go back to Step2.

3.3.4 Armijo Method

We have used the modified Newton method as our descent method for the optimization problem. In this method, we use the Armijo algorithm [23] to determine the step size in the descent direction. Its steps are as follows:

Step 1: The parameters of the Armijo method are initialized as follows:

$$\beta \in \left[\frac{1}{10}; \frac{1}{2} \right], (x_a^0, C_a^0), \sigma \in [10^{-5}; 10^{-1}], m = 0 \text{ and } t = 0.$$

Step 2: The capacity and flow of an arc are calculated for the next iteration using:

$$(C_a^{t+1}, x_a^{t+1}) = (C_a^t, x_a^t) + \sigma \beta^m d_M^t$$

Step 3: If $L(C_a^t, x_a^t, \mu_a^t) - L(C_a^{t+1}, x_a^{t+1}, \mu_a^t) \geq -\sigma \beta^m \nabla L(C_a^t, x_a^t, \mu_a^t) d_M^t$ Stop and return β^m (the step size is returned to the modified Newton method).

Else $m := m + 1$ and go back to Step 2.

3.4 SALA Applied to the Global Reconfiguration strategy

Before presenting the SALA algorithm for the global reconfiguration strategy, we first introduce some related parameters.

Let, u be Lagrange multipliers defined in our strategy as $u = (u_1, \dots, u_n, U_1, U_2, \dots, U_k)$

and we have considered two parameters δ_a^t and δ_k^t for the feasibility of the SALA based global

restoration strategy as: $\delta_a^t = (x_a^{st} - \sum_{k=1}^K \sum_{p=1}^{N_k} \pi_{kp}^a x_{kp}^{st})$ and $\delta_k^t = (\sum_p x_{kp}^{st} - r_k)$. A current solution

provided by the global reconfiguration strategy is feasible if $\forall a, \delta_a^t = 0, 0 \leq x_a^{st} < C_a^{st}$ and

$$\forall k, p, \delta_k^t = 0 \text{ and } x_{kp}^{st} \geq 0.$$

The optimality of the proposed optimization model for the global restoration strategy is obtained from the Karush-Kuhn_Tucker (KKT) [12] conditions. In the SALA based global reconfiguration strategy, an optimal solution must verify the following constraints

$$\frac{\gamma_a}{C_a^{s^*} - x_a^{s^*}} + d_a = u_a^* \text{ if } x_a^{s^*} > 0 \quad (3.35)$$

$$\frac{\gamma_a}{C_a^{s^*} - x_a^{s^*}} + d_a \geq u_a^* \text{ if } x_a^{s^*} = 0 \quad (3.36)$$

$$\sum_a \pi_{kp}^a u_a^* = U_k^* \text{ if } x_{kp}^{s^*} > 0 \quad (3.37)$$

$$\sum_a \pi_{kp}^a u_a^* > U_k^* \text{ if } x_{kp}^{s^*} = 0 \quad (3.38)$$

$$0 \leq x_a^{st} < C_a^{s^*} \quad (3.39)$$

$$x_{kp}^{s^*} \geq 0 \quad (3.40)$$

And the KKT conditions can be defined as:

$$\sum_a \pi_{kp}^a \left(\frac{\gamma_a}{C_a^{s^*} - x_a^{s^*}} + d_a \right) = U_k^* \text{ if } x_{kp}^{s^*} > 0 \quad (3.41)$$

$$\sum_a \pi_{kp}^a \left(\frac{\gamma_a}{C_a^{s^*} - x_a^{s^*}} + d_a \right) > U_k^* \text{ if } x_{kp}^{s^*} = 0 \quad (3.42)$$

It can be observed that in the SALA based global reconfiguration, the optimal paths are the shortest paths where the arc costs are given by the cost function $\phi_a(x_a^s, C_a^s) = \frac{\gamma_a}{C_a^{s^*} - x_a^{s^*}} + d_a$ which can be interpreted as the delay transit cost and the installation cost of a unit of capacity on arc a .

Finally, we can set a feasible solution is optimal if the parameter $\sigma_a = 0$ and the parameter $\sigma_{kp} = 0$ where,

$$\sigma_a = \phi_a(x_a^s, C_a^s) - u_a \text{ if } x_a^s > 0 \text{ or } \sigma_a = \min\{\phi_a(x_a^s, C_a^s) - u_a, 0\} \text{ if } x_a^s = 0 \text{ and}$$

$$\sigma_{kp} = \left(\sum_a \pi_{kp}^a(u_a) - U_k\right) \text{ if } x_{kp}^s > 0 \text{ or } \sigma_{kp} = \min\left\{\left(\sum_a \pi_{kp}^a(u_a) - U_k\right), 0\right\} \text{ if } x_{kp}^s = 0$$

The SALA algorithm for the global reconfiguration strategy can then be described as follows:

SALA for Global Reconfiguration Strategy

Step1: Initialize all the necessary parameter values as follows:

$\varepsilon > 0, u_1^1, \dots, u_n^1, \lambda_0, \beta, x_1^0, \dots, x_n^0, C_1^0, \dots, C_n^0, x_{11}^{s_0}, \dots, x_{k1}^{s_0}, \Delta_1^0, \dots, \Delta_n^0, n_1^1, \dots, n_n^1, t$, Where u_a^1 is the Lagrange multipliers, λ_0 is the penalty parameter, $x_{11}^{s_0}, \dots, x_{k1}^{s_0}$ are the initial flows of all the working paths of all the commodities, x_1^0, \dots, x_n^0 and C_1^0, \dots, C_n^0 are the initial flows and capacities of the all arcs, t iteration number, $n_1^1 - 1$ is the initial total number of paths through the arcs etc.

Step 2: For all arcs $a = 1, 2, \dots, n$, Calculate

$$(C_a^{st}, x_a^{st}) = \arg_{0 \leq x_a^s < C_a} \min \left\{ \frac{\gamma_a x_a^s}{C_a^s - x_a^s} + d_a C_a^s + u_a^t x_a^s + \frac{\lambda_t}{2} \left[(x_a^s)^2 - 2x_a^s (x_a^{s(t-1)}) - \frac{\delta_a^{t-1}}{n_a^t} \right] \right\}$$

This is done by using the modified Newton and Armijo method.

Step 3: End (For) of step 2.

Step 4: For all $k = 1, 2, \dots, K$, Compute

$$\theta = \min_{p=1, \dots, N_k} \left\{ \sum_{a=1}^n \pi_{kp}^a \phi_a(x_a^{st}, C_a^{st}) \right\}$$

Then determine η , the cost of a new previously unexplored shortest path where the arc costs are equal to $\phi_a(x_a^{st}, C_a^{st})$.

Step 5: If $\eta < \theta$, then introduce a new working path for the commodity i.e. $N_k = N_k + 1$.

Step 6: End (If) of step 5.

Step 7: While $p \leq N_k$, Compute

$$x_{kp}^{st} = \max \left\{ 0; x_{kp}^{st-1} + \frac{(\sum_{a=1}^n \pi_{kp}^a u_a^t) - U_k^t}{\lambda_t (1 + \sum_{a=1}^n \pi_{kp}^a)} + \frac{\sum_{a=1}^n \left(\pi_{kp}^a \frac{\delta_a^{t-1}}{n_a^t} \right) - \frac{\delta_k^{t-1}}{N_k}}{1 + \sum_{a=1}^n \pi_{kp}^a} \right\}$$

Step 8: If $x_{kp}^{st} = 0$ $N_k = N_k - 1$ (Cancellation of a non active working path).

Step 9: End (If) of step 8.

Step 10: End (While) of step 7.

Step 11: End (For) of step 4.

Step 12: Compute $\delta^t = \max \left\{ \max_a |\delta_a^t|, \max_k |\delta_k^t| \right\}$ and $\sigma^t = \max \left\{ \max_a |\sigma_a^t|, \max_{kp} |\sigma_{kp}^t| \right\}$

Step 13: If $\max \{ \delta^t, \sigma^t \} < \varepsilon$ Stop.

Step 14: End (If) of step 13.

Step 15: Else go to step 16.

Step 16: Update the parameters using the following formulas:

$$u_a^{t+1} = u_a^t + \frac{\lambda_t}{n_a^t} \delta_a^t$$

$$U_k^{t+1} = U_k^t + \frac{\lambda_t}{N_k} \delta_k^t$$

$$n_a^{t+1} = 1 + \sum_{k=1}^K \sum_{p=1}^{N_k} \pi_{kp}^a$$

$$\lambda_{t+1} = \beta \lambda_t$$

$t \leftarrow t + 1$ and go back to Step 2.

3.5 SALA Applied to the Failure-Oriented Reconfiguration

Strategy

Before presenting the SALA algorithm for the failure-oriented reconfiguration strategy, we first introduce some related parameters.

The Lagrange multiplier u is considered as $u^T = (u_1, u_2, \dots, u_n, U_1, \dots, U_k, V_1, \dots, V_k)$. Let us consider the following notations, at iteration t

$$m_a^t = 1 + \sum_{k=1}^K \left[\sum_{p=1}^{N_k} (1 - \mu_{kp}^s) \pi_{kp}^a + \sum_{q=1}^{M_k} \alpha_{kq}^a \right] \quad (3.43)$$

In Equation (3.43), $(m_a^t - 1)$ represents the total number of paths, both primary and back up paths for all commodities passing through link a .

$$\Delta_a^t = z_a^{st} - \sum_{k=1}^K \left[\sum_{p=1}^{N_k} (1 - \mu_{kp}^s) \pi_{kp}^a x_{kp}^{s_0^t} + \sum_{q=1}^{M_k} \alpha_{kq}^a y_{kq}^{ts} \right] \quad (3.44)$$

$$\nabla_k^t = \sum_{p=1}^{N_K} x_{kp}^{s_0^t} - r_k \quad (3.45)$$

$$D_k^t = \sum_{p=1}^{N_K} \mu_{kp}^s x_{kp}^{s_0^t} - \sum_{q=1}^{M_K} y_{kq}^{st} \quad (3.46)$$

Feasible conditions to the SALA based failure-oriented reconfiguration strategy are obtained when $\Delta_a^t = 0$, $\nabla_k^t = 0$ and $D_k^t = 0$. It should be noted that (3.44) ensures that the aggregated flow constraints for each link are maintained, (3.45) assures that the bandwidth requirement constraints of the commodities are met and (3.46) ensures that all the affected traffic flows are rerouted in case of a link failure.

The optimality conditions in the SALA based failure-oriented reconfiguration come from the KKT conditions [12] as follows.

Let $(C_a^{s^*}, z_a^{s^*}, x_{kp}^{s_0^*}, y_{kq}^{s^*})$ be an optimal solution provided by the restoration strategy and $(u_1^*, \dots, u_n^*, U_1^*, \dots, U_k^*, V_1^*, \dots, V_k^*)$ be the associated Lagrange multipliers; the KKT conditions involve the following Equations

$$\phi_a(z_a^{s^*}, C_a^{s^*}) = u_a^* \text{ if } z_a^{s^*} > 0 \quad (3.47)$$

$$\phi_a(z_a^{s^*}, C_a^{s^*}) > u_a^* \text{ if } z_a^{s^*} = 0 \quad (3.48)$$

$$\sum_a \pi_{kp}^a u_a^* (1 - \mu_{kp}^s) = U_k^* - \mu_{kp}^s V_k^* \text{ if } x_{kp}^{s_0^*} > 0 \quad (3.49)$$

$$\sum_a \pi_{kp}^a u_a^* (1 - \mu_{kp}^s) > U_k^* - \mu_{kp}^s V_k^* \text{ if } x_{kp}^{s_0^*} = 0 \quad (3.50)$$

$$\sum_a \alpha_{kq}^a u_a^* = V_k^* \text{ if } y_{kq}^{s*} > 0 \quad (3.51)$$

$$\sum_a \alpha_{kq}^a u_a^* > V_k^* \text{ if } y_{kq}^{s*} = 0 \quad (3.52)$$

We observe that, at the optimum, the primary and the backup paths for the commodities are the shortest paths where the arc costs are given by the function $\phi_a(z_a^s, C_a^s)$. Here, the primary paths lengths are equal to $(U_k^* - \mu_{kp}^s V_k^*)$ for commodity k and the backup paths lengths are equal to V_k^* . We have considered three parameters Γ_a^t , Γ_{kp}^t and Ω_{kq}^t for the optimality of the failure-oriented reconfiguration strategy. The parameters Γ_a^t , Γ_{kp}^t and Ω_{kq}^t are defined as follows:

$$\Gamma_a^t = \phi_a(z_a^{st}, C_a^{st}) - u_a^t \text{ if } z_a^s > 0 \quad (3.53)$$

$$\Gamma_a^t = \min\{\phi_a(z_a^{st}, C_a^{st}) - u_a^t; 0\} \text{ if } z_a^s = 0 \quad (3.54)$$

$$\Gamma_{kp}^t = (1 - \mu_{kp}^s)(\sum_a \pi_{kp}^a u_a^t) - U_k^t - \mu_{kp}^s V_k^t \text{ if } x_{kp}^{s0t} > 0 \quad (3.55)$$

$$\Gamma_{kp}^t = \min\{(1 - \mu_{kp}^s)(\sum_a \pi_{kp}^a u_a^t) - U_k^t - \mu_{kp}^s V_k^t; 0\} \text{ if } x_{kp}^{s0t} = 0 \quad (3.56)$$

$$\Omega_{kq}^t = (\sum_a \alpha_{kq}^a u_a^t) - V_k^t \text{ if } y_{kq}^s > 0 \quad (3.57)$$

$$\Omega_{kq}^t = \min\{(\sum_a \alpha_{kq}^a u_a^t) - V_k^t; 0\} \text{ if } y_{kq}^s = 0 \quad (3.58)$$

From the above (3.53) to (3.58) and KKT conditions, a solution provided by the SALA based failure-oriented reconfiguration strategy is optimal if and only if $\Gamma_a^t = \Gamma_{kp}^t = \Omega_{kq}^t = 0$. The algorithm for the SALA based failure-oriented reconfiguration strategy can then be discussed as follows:

Step1: Initialize all the necessary parameter values as follows:

$\varepsilon > 0, u_1^1, \dots, u_n^1, U_1, U_2, \dots, U_k, \lambda_0, \beta, x_1^0, \dots, x_n^0, C_1^0, \dots, C_n^0, x_{11}^{s_0}, \dots, x_{k1}^{s_0}, y_{11}^s, \dots, y_{k1}^s, \Delta_1^0, \dots, \Delta_n^0$
 $m_1^1, \dots, m_n^1, t, D_1^0, \dots, D_k^0$ Where u_1^1 and U_1 are the Lagrange multipliers, λ_0 is the penalty parameter, $x_{11}^{s_0}, \dots, x_{k1}^{s_0}$ are the initial flows of all the working paths of all the commodities, $y_{11}^s, \dots, y_{k1}^s$, are the initial flows of the backup paths, z_1^0, \dots, z_n^0 and C_1^0, \dots, C_n^0 are the initial flows and capacities of the all arcs, t iteration number, $m_1^1 - 1$ is the initial total number of paths through each of the arc etc.

Step 2: For all arcs $a = 1, 2, \dots, n$, Calculate

$$(z_a^{st}, C_a^{st}) = \arg_{0 \leq z_a^s < C_a} \min \left\{ \frac{\gamma_a z_a^s}{C_a^s - z_a^s} + d_a C_a^s + u_a^t z_a^s + \frac{\lambda_t}{2} \left[(z_a^s)^2 - 2z_a^s (z_a^{s(t-1)} - \frac{\Delta_a^{t-1}}{m_a^t}) \right] \right\}$$

This is achieved by using the Modified-Newton method and Armijo method.

Step 3: End (For) of step 2.

Step 4: For all $k = 1, 2, \dots, K$, Compute

$$\theta_1 = \min_{\{p=1, \dots, N_k / \mu_{kp}^s = 0\}} \left\{ \sum_{a=1}^n (1 - \mu_{kp}^s) \tau_{kp}^a \phi_a(z_a^{st}, C_a^{st}) \right\}$$

Then, determine η_1 , the cost of a new previously unexplored shortest path, where the arc costs are equal to $\phi_a(z_a^{st}, C_a^{st})$.

Step 5: If $\eta_1 < \theta_1$, then $N_k = N_k + 1$ (A new working path for the commodity is introduced).

Step 6: End (If) of step 5.

Step 7: While $p \leq N_k$ compute

$$x_{kp}^{s_0^t} = \max \left\{ 0; x_{kp}^{s_0^{(t-1)}} + \frac{(1 - \mu_{kp}^s)(\sum_{a=1}^n \pi_{kp}^a u_a^t) - U_k^t + \mu_{kp}^s V_k}{\lambda_t(1 + \mu_{kp}^s + (1 - \mu_{kp}^s)\sum_{a=1}^n \pi_{kp}^a)} + \frac{(1 - \mu_{kp}^s)\left(\sum_{a=1}^n \pi_{kp}^a \frac{\Delta_a^{t-1}}{m_a^t}\right) - \frac{\nabla_k^{t-1}}{N_k}}{1 + \mu_{kp}^s + (1 - \mu_{kp}^s)\sum_{a=1}^n \pi_{kp}^a} \right\}$$

Step 8: If $x_{kp}^{s_0^t} = 0$ $N_k = N_k - 1$ (Cancellation of a non active working path).

Step 9: End (If) of step 8.

Step 10: Else If $\mu_{kp}^s = 1$ Compute

$$\theta_2 = \min_{q=1, \dots, M_k} \left\{ \sum_{a=1}^n \alpha_{kq}^a \phi(z_a^{st}, C_a^{st}) \right\}$$

Then, determine η_2 the length of the new shortest backup path where the arc costs are equal to

$$\phi_a(z_a^{st}, C_a^{st}).$$

Step 11: If $\eta_2 < \theta_2$, then $M_k = M_k + 1$ (Introduction of a new backup path).

Step 12: End (If) of step 11.

Step 13: While $q \leq M_k$ Compute

$$y_{kq}^{st} = \max \left\{ 0; y_{kq}^{s^{(t-1)}} + \frac{V_k^t + \sum_{a=1}^n \alpha_{kq}^a u_a}{\lambda_t(1 + \sum_{a=1}^n \alpha_{kq}^a)} + \frac{\frac{D_k^t}{N_k + M_k} + \sum_{a=1}^n \alpha_{kq}^a \frac{\Delta_a^{t-1}}{m_a^t}}{1 + \sum_{a=1}^n \alpha_{kq}^a} \right\}$$

Step 14: If $y_{kq}^{st} = 0$ then $M_k = M_k - 1$ (Cancellation of a non active backup path)

Step 15: End (If) of step 14.

Step 16: End (While q) of step 13.

Step 17: End (Else) of step 10.

Step 18: End (While p) of step 7.

Step 19: End (For k) of step 4.

Step 20: Compute

$$R^t = \max \left\{ \max_a |\Delta'_a|, \max_k |\nabla'_k|, \max_k |D'_k| \right\} \text{ and } G^t = \max \left\{ \max_a |\Gamma'_a|, \max_{kp} |\Gamma'_{kp}|, \max_{kq} |\Omega'_{kq}| \right\}$$

Step 21: If $\max\{R^t, G^t\} < \varepsilon$, then stop.

Step 22: End (If) of step 21.

Step 23: Else, go to Step 24.

Step 24: Update the parameters using the following formulas:

$$u_a^{t+1} = u_a^t + \frac{\lambda_t}{m_a^t} \Delta'_a$$

$$U_k^{t+1} = U_k^t + \frac{\lambda_t}{N_k} \nabla'_k$$

$$V_k^{t+1} = V_k^t + \frac{\lambda_t}{N_k + M_k} D'_k$$

$$m_a^{t+1} = 1 + \sum_{k=1}^K \left[\sum_{p=1}^{N_k} \pi_{kp}^a (1 - \mu_{kp}^s) + \sum_{q=1}^{M_k} \alpha_{kq}^a \right]$$

$$\lambda_{t+1} = \beta \lambda_t$$

$t \leftarrow t + 1$ and go back to Step 2.

We have described our proposed two nonlinear optimization models for the global reconfiguration strategy and the failure-oriented reconfiguration strategy in this chapter. We have also described our solution approaches using the SALA algorithms.

Chapter 4

Performance Evaluation

This chapter presents the numerical results of the global reconfiguration strategy versus the failure-oriented reconfiguration strategy for designing self-healing ATM networks. These two restoration strategies are evaluated quantitatively in terms of five performance metrics which are: capacity installation cost, routing cost, total network cost (i.e. the summation of capacity installation cost and routing cost), total used capacity for a network to optimize the flow and capacity and the CPU time required by the algorithms to yield an optimal solution. Different network scenarios, network topologies, under both symmetric and asymmetric traffic demands for the commodities (source-destination pairs) are considered.

4.1 Parameters for Global and Failure-Oriented Reconfiguration Strategies:

Several parameters are considered for evaluating the performances of the global and failure-oriented reconfiguration strategies. These are summarized as follows:

4.1.1 Network Topologies

With respect to size, we have considered small, medium and large size networks such as: network with 3 nodes, 6 arcs denoted as Network 1, network with 5 nodes, 20 arcs denoted as Network 2, network with 8 nodes, 56 arcs denoted as Network 3, network with 12 nodes, 126 arcs denoted as Network 4 and network with 15 nodes, 162 arcs denoted as Network 5.

With respect to connectivity, we have considered four networks of 15 nodes with 132, 142, 152 and 162 arcs which are denoted as N (15,132), N (15,142), N (15,152) and N (15,162).

4.1.2 Traffic Demands

Both symmetric and asymmetric traffic demands are considered. In case of symmetric traffic demands, a fixed bandwidth requirement (5 bits per second) is assigned for all the commodities of the networks. For asymmetric traffic demands, the bandwidth requirements for the commodities are generated randomly within a fixed range using the following function:

```

double randdouble( double min, double max)
{
    double result;
    if (min>max)
        result = max + (rand()/((( double)RAND_MAX + 1) / min));
    else result = min + (rand()/((( double)RAND_MAX + 1) / max));
    return result;
}

```

The above mentioned function generates random numbers within the range defined by its two parameters min and max.

4.1.3 Network Scenarios

We have considered three different network scenarios which are as follows:

- Scenario 1: In this scenario, network connectivity is varied i.e. networks have the same number of nodes but with different number of arcs. Both the symmetric and asymmetric traffic demands are considered.

- Scenario 2: The network size is varied i.e. we have taken small, medium and large size networks. Both symmetric and asymmetric traffic demands are considered.
- Scenario 3: Different traffic demands are considered for the commodities and the network size remains fixed.

4.1.4 Performance Metrics

The performance metrics used for comparing the global and failure-oriented reconfiguration strategies are described as follows.

4.1.4.1 Capacity Installation Cost

Capacity installation cost is the cost of the installed capacities on the arcs of the network. In other words, capacity installation cost is the amount of money which is needed to install the links or arcs of a network. To simplify the network design problem, we have assumed that all the arcs of the networks have the same unit capacity installation cost.

4.1.4.2 Total Used Capacity

This metric defines the required capacity which is needed to minimize the total network cost while fulfilling all traffic constraints. In other words, the total used capacity is the summation of the capacities of all arcs of the obtained optimal solution.

4.1.4.3 Routing Cost

The routing cost defines the cost generated when the traffic gets routed in an arc of the network. For simplicity, it has been assumed that all arcs in networks have same unit routing cost.

4.1.4.4 Total Network Cost

The total network cost is the aggregation of capacity installation cost and routing cost of the network. In fact, total network cost is the main objective function of our CFA problems which we want to minimize while maintaining all traffic requirements.

4.1.4.5 CPU Time

The CPU time defines the time required to execute the restoration algorithms. A restoration strategy should be fast enough to work in a real life situation. To quickly restore affected traffic upon network failure, a restoration strategy should reroute the affected traffics within two seconds [24].

We have run our algorithms in an Intel (R) Xeon (R) processor at the speed of 1.86GHz with 1 GB memory.

Traffic requirement parameters are captured in Table 4.1.

Table 4.1: Summary of all parameters for Global and Failure-Oriented Reconfiguration Strategies

Considered Networks(with respect to size)	Network 1, Network 2, Network 3, Network 4 and Network 5.
Considered Networks (with respect to connectivity)	N (15,132), N (15,142), N (15,152) and N (15,162).
Network scenarios	Scenario 1, Scenario 2 and Scenario 3.
Candidate paths per commodity	All possible paths
Maximum number of paths per node pair	10
Maximum hop limit of paths	10
Traffic demands	Both symmetric and asymmetric
Link orientation	Bidirectional
Failure scenarios	Single link failure scenario

<p>Initial parameter values for the SALA based Global and Failure-Oriented Reconfiguration algorithms</p>	<p>Global reconfiguration: $\gamma_a = .001$ (unit routing cost), $d_a = .001$ (unit capacity installation cost), $u_a = .001$ for N(3,6), N(5,20), N(8,56) and N(12,126), $u_a = .0005$ for N(15,162), $U_k = .002$ for N(5,20), N(8,56) and N(12,126) and $U_k = .0007$ for N(15,162) where u_a and U_k are Lagrange multipliers, $\beta = 1.5$ for N(5,20), N(8,56) and N(12,126) and $\beta = 1.3$ (needed to update λ) for N(15,162), $\lambda = .001$ (penalty parameter).</p>
	<p>Failure-oriented: $\gamma_a = .001$, $d_a = .001$, $u_a = .001$, $U_k = .001$, $V_k = .001$ (Lagrange multiplier), $\beta = 2$, $\lambda = .0001$</p>
<p>Performance metrics</p>	<p>Capacity Installation Cost, Routing Cost, Total Network Cost, Total Used capacity, and CPU Time</p>
<p>Processor configuration</p>	<p>Processor: Intel Xeon 1.86 GHz with Memory: 1 GB</p>
<p>Restoration level</p>	<p>100% restoration for the predefined failure scenario</p>
<p>Capacity modularization</p>	<p>Not considered</p>

4.2 Comparison of Global Reconfiguration Strategy and Failure-Oriented Reconfiguration Strategy

4.2.1 Results Using Scenario 1

4.2.1.1 Capacity Installation Cost

Here, the considered four networks are: N(15,132), N(15,142), N(15,152) and N(15,162).

The results for capacity installation cost using symmetric traffic demands are captured in Table 4.2 and depicted in Figure 4.1.

Table 4.2: Capacity Installation Cost (in Dollars) for Global vs. Failure-Oriented Reconfiguration Strategy using symmetric traffic demands

Networks	N(15, 132)	N(15, 142)	N(15, 152)	N(15, 162)
Global	1.8414	1.7741	1.702	1.623
Failure-Oriented	4.359	4.209	4.059	3.989

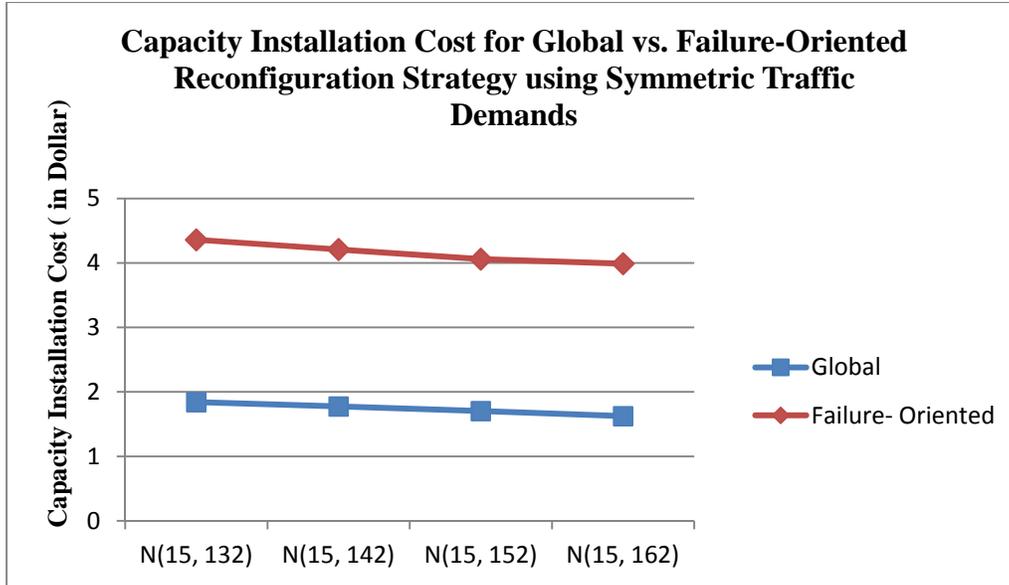


Figure 4.1: Capacity Installation Cost for Global vs. Failure-Oriented Reconfiguration Strategy using symmetric traffic demands

In Figure 4.1, it can be observed that the global reconfiguration strategy generates less capacity installation cost than the failure-oriented reconfiguration strategy when using symmetric traffic demands. This can be attributed to the fact that in failure-oriented reconfiguration, only the traffics of the affected working paths are rerouted, whereas in global reconfiguration, the whole layout of working paths (affected and unaffected) may be rearranged to overcome a failed link. Due to this, the global reconfiguration optimization procedure is more flexible in selecting candidate paths and thus better shares the capacity of the network than the failure-oriented optimization procedure.

In Figure 4.1, it can also be observed that, as the number of arcs increases, the capacity installation cost decreases. This can be justified by the fact that with the increment in the number of arcs, better options for candidate paths are available for each commodity when using both

reconfiguration strategies, which explains the progressive decrease in capacity installation cost since less capacity installation cost would be required.

One more interesting phenomenon revealed from Figure 4.1 is that, the difference with respect to capacity installation cost between the global and failure-oriented reconfiguration strategies is more pronounced in a network having fewer arcs.

Likewise symmetric traffic demand, we have also compared the restoration strategies with respect to capacity installation cost under asymmetric traffic demands. Table 4.3 summarizes the obtained results, which are depicted in Figure 4.2.

Table 4.3: Capacity Installation Cost (in Dollars) of Global vs. Failure-Oriented Reconfiguration Strategy using asymmetric traffic demands

Networks	N(15, 132)	N(15, 142)	N(15, 152)	N(15, 162)
Global	0.751	0.725	0.696	0.668
Failure-Oriented	1.743	1.683	1.623	1.595

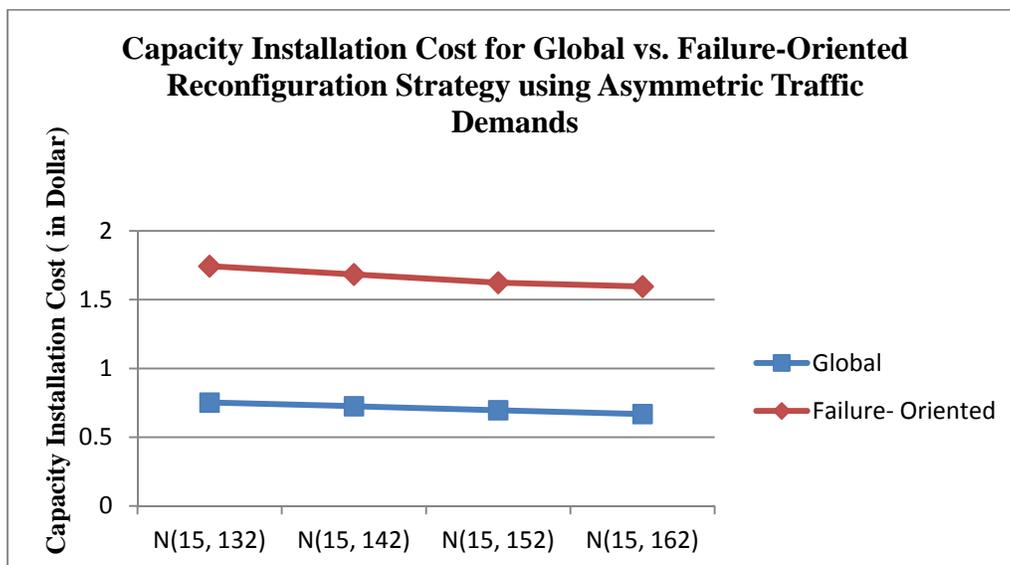


Figure 4.2: Capacity Installation Cost of Global vs. Failure-Oriented Reconfiguration Strategy using asymmetric traffic demands

In Figure 4.2, a similar trend that is observed in the case of symmetric traffic demands prevails for both the reconfiguration strategies, i.e., global reconfiguration is still better than failure oriented reconfiguration in terms of capacity installation cost due to the fact that global reconfiguration requires less capacity than failure-oriented and the difference between both strategies is less pronounced in a network having large number of arcs.

4.2.1.2 Total Used Capacity

The results obtained when evaluating this performance metric are summarized in Table 4.4 and depicted in Figure 4.3.

Table 4.4: Total Used Capacity (in Bits) for Global vs. Failure-Oriented Reconfiguration Strategy using symmetric traffic demands

Networks	N(15, 132)	N(15, 142)	N(15, 152)	N(15, 162)
Global	1841.42	1774.16	1702.8	1623.62
Failure-Oriented	4359.84	4209.83	4059.81	3989.77

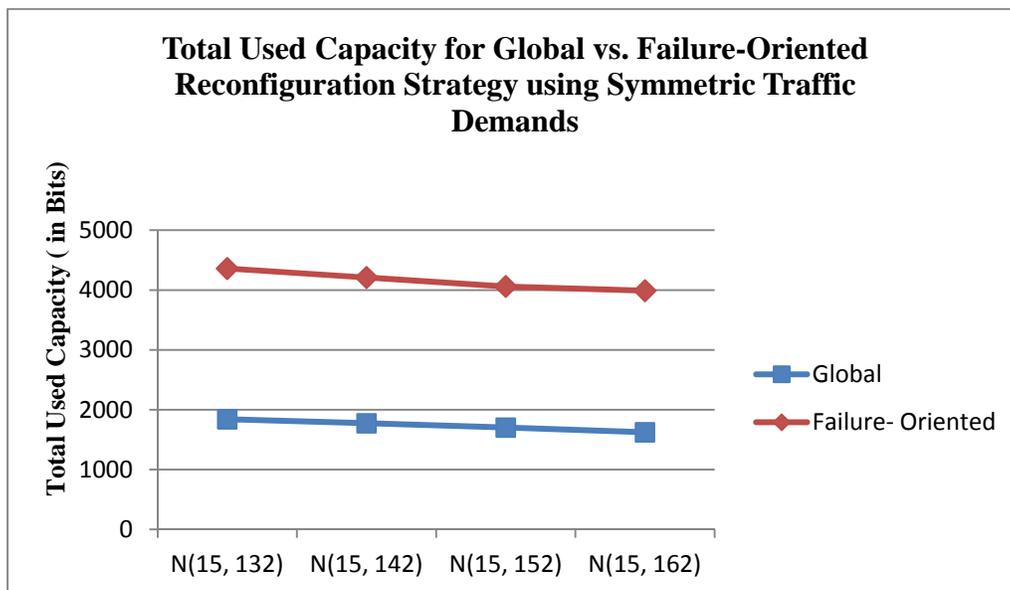


Figure 4.3: Total Used Capacity for Global vs. Failure-Oriented Reconfiguration Strategy using symmetric traffic demands

Figure 4.3 shows that global reconfiguration requires less capacity than failure-oriented reconfiguration in the case of symmetric traffic demands. This is due to the fact that, in global reconfiguration, all the working paths are rearranged whether they are affected or unaffected, in order to overcome a failed link whereas in the failure-oriented case, only the affected paths are reconfigured to restore the affected traffic in the network. As a result, the global reconfiguration optimization procedure is more flexible than the failure-oriented optimization procedure with respect to capacity sharing when a failure occurs. It can also be observed in Figure 4.3 that the difference between global and failure-oriented reconfiguration strategy in terms of required capacity is less pronounced in networks with large number of arcs compared to networks with fewer number of arcs.

In case of asymmetric traffic demand, the results for the total used capacity are captured in Table 4.5 and depicted in Figure 4.4.

Table 4.5: Total Used Capacity (in Bits) for Global vs. Failure-Oriented Reconfiguration Strategy using asymmetric traffic demands

Networks	N(15, 132)	N(15, 142)	N(15, 152)	N(15, 162)
Global	751.606	725.361	696.066	668.39
Failure-Oriented	1743.89	1683.89	1623.88	1595.92

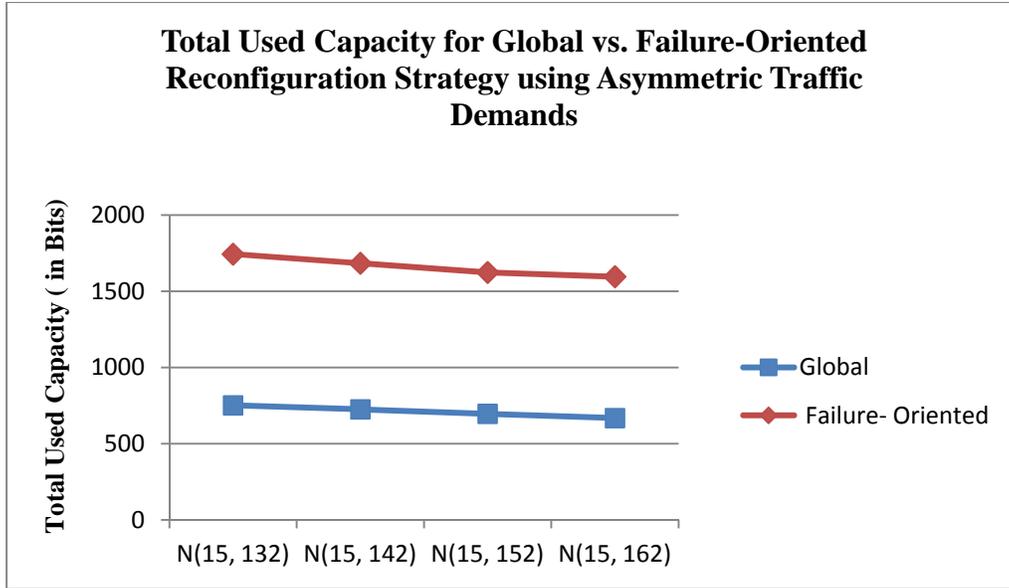


Figure 4.4: Total Used Capacity for Global vs. Failure-Oriented Reconfiguration Strategy using asymmetric traffic demands

In Figure 4.4, it can be observed that, the two phenomena observing in the case of symmetric traffic are repeated for asymmetric traffic demands i.e. when the number of arc increases, the total used capacity decreases for both restoration strategies and global reconfiguration results in lower required capacity compared to failure-oriented reconfiguration.

4.2.1.3 Routing Cost

The results for this metric are summarized in Table 4.6 and depicted in Figure 4.5.

Table 4.6: Routing Cost (in Dollars) for Global vs. Failure-Oriented Reconfiguration Strategy using symmetric traffic demands

Networks	N(15, 132)	N(15, 142)	N(15, 152)	N(15, 162)
Global	0.643	0.695	0.749	0.783
Failure-Oriented	0.066	0.071	0.076	0.082

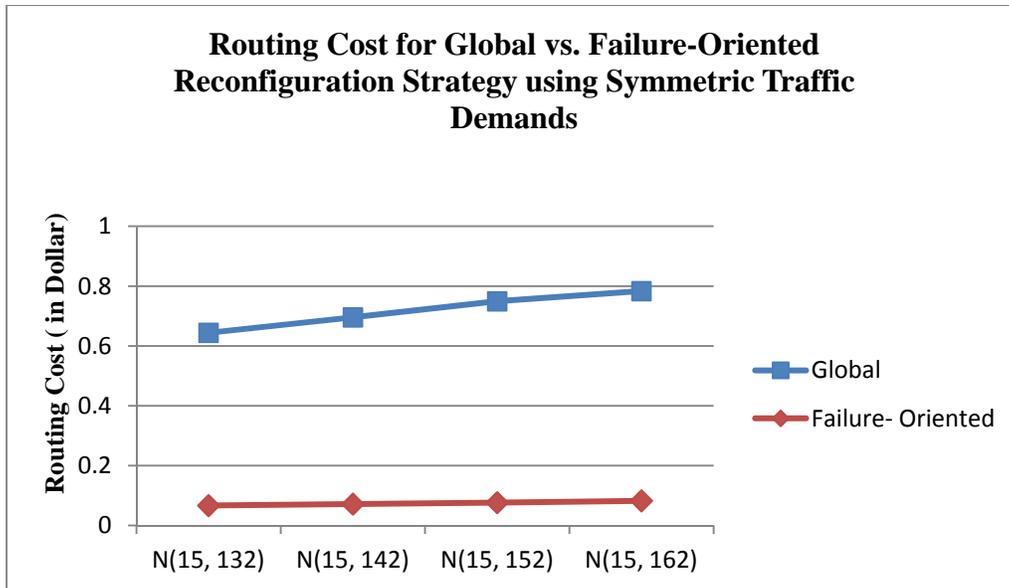


Figure 4.5: Routing Cost for Global vs. Failure-Oriented Reconfiguration Strategy using symmetric traffic demands

In Figure 4.5, it can be observed that global reconfiguration generates more routing cost than the failure-oriented reconfiguration. This is due to the fact that, in global reconfiguration, in case of a link failure, all traffic flows (affected and unaffected) are rerouted so that the reconfiguration ratio can be guaranteed but the total network cost is minimized. However, in the failure-oriented reconfiguration, only the affected traffic flows are rerouted when failure occurs i.e. unaffected traffic flows remain unchanged.

It can also be observed in Figure 4.5 that when the number of arcs is increased gradually, the routing cost is also gradually increased for both restoration strategies. This might be due to the fact that, with the increment in the number of arcs, the restoration strategies get better options of selecting the candidate routes whereas the routing cost becomes larger. Lastly, one more interesting phenomenon can also be observed that, the difference between the restoration strategies in terms of routing cost is more prominent in network with large number of arcs.

In case of asymmetric traffic demands, the comparison results are summarized in Table 4.7 and depicted in Figure 4.6.

Table 4.7: Routing Cost (in Dollars) for Global vs. Failure-Oriented Reconfiguration Strategy using asymmetric traffic demands

Networks	N(15, 132)	N(15, 142)	N(15, 152)	N(15, 162)
Global	0.499	0.526	0.551	0.567
Failure-Oriented	0.066	0.071	0.076	0.082

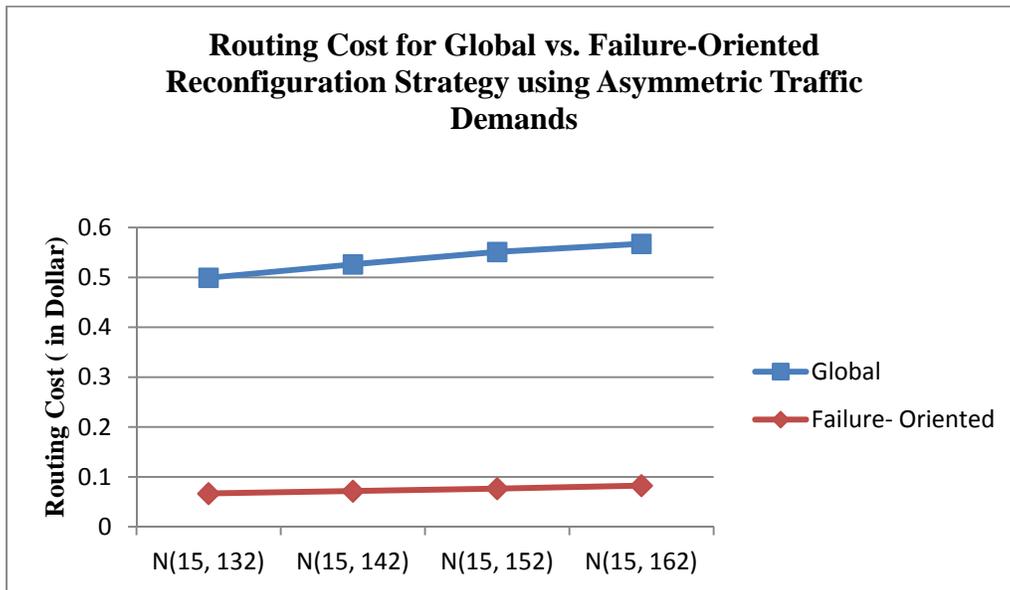


Figure 4.6: Routing Cost for Global vs. Failure-Oriented Reconfiguration Strategy using asymmetric traffic demands

In Figure 4.6, it can be observed that the failure-oriented reconfiguration yields less routing cost compared to the global reconfiguration for asymmetric traffic demands due to the fact that the global reconfiguration strategy reroutes all affected and unaffected traffics for failure recovery. Moreover, the difference between the two strategies in terms of routing cost is more pronounced in network with large number of arcs. This might be due to the fact that the

difference with respect to routing cost is increased, when there are more available routes for routing the traffics.

4.2.1.4 Total Network Cost

Table 4.8 summarizes the total network cost for global and failure-oriented reconfiguration strategies under symmetric traffic demands.

Table 4.8: Total Network Cost (in Dollars) for Global vs. Failure-Oriented Reconfiguration Strategy using symmetric traffic demands

Networks	N(15, 132)	N(15, 142)	N(15, 152)	N(15, 162)
Global	2.485	2.469	2.452	2.406
Failure-Oriented	4.426	4.281	4.136	4.072

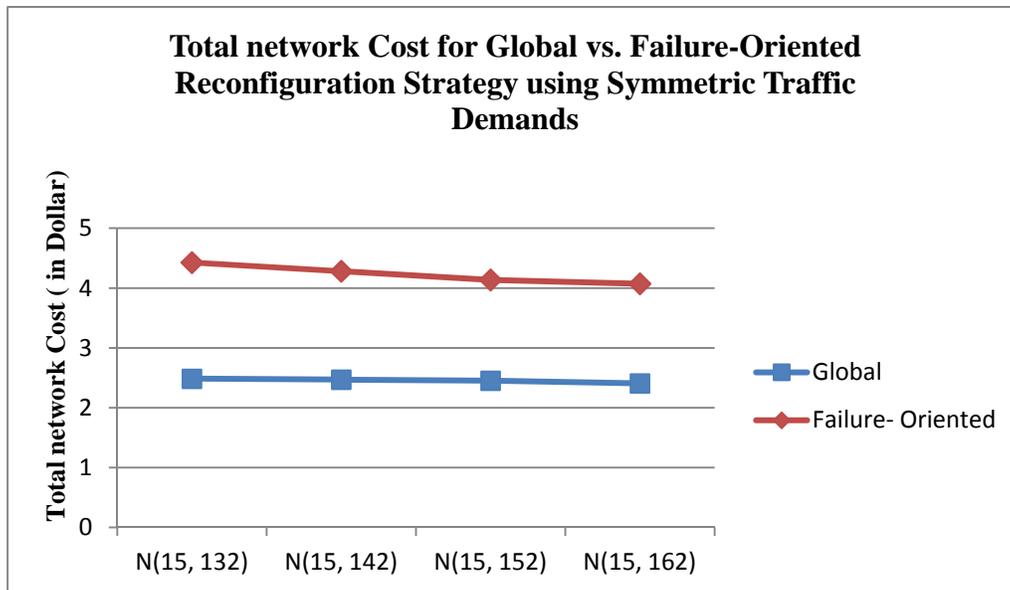


Figure 4.7: Total Network Cost for Global vs. Failure-Oriented Reconfiguration Strategy using symmetric traffic demands

In Figure 4.7, it can be observed that when the number of arcs is gradually increasing, the total network cost is gradually decreasing for both restoration strategies. This can be justified by the fact that in the calculation of the total network cost, the capacity installation cost has larger value than the routing cost. Thus, the trend of the total network cost inherits from the trend of capacity installation cost. It is already revealed that, with the increase of the number of arcs, the capacity installation cost decreases and as a result the total network cost also decreases for both restoration strategies.

In Figure 4.7, it can also be seen that global reconfiguration strategy generates less network cost than failure-oriented reconfiguration when optimizing the network. This observation is understandable as the global reconfiguration optimization procedure is more flexible than the failure-oriented one in selecting the restoration routes. So, global optimization procedure requires less capacity which in turn, results in less capacity installation cost and total network cost.

For asymmetric traffic demands, the same trend that is observed for symmetric traffic demands prevails. The results of the performance metric total network cost are summarized in Table 4.9 and depicted in Figure 4.8.

Table 4.9: Total Network Cost (in Dollars) for Global vs. Failure-Oriented Reconfiguration Strategy using asymmetric traffic demands

Networks	N(15, 132)	N(15, 142)	N(15, 152)	N(15, 162)
Global	1.251	1.25	1.247	1.235
Failure-Oriented	1.81	1.755	1.7	1.678

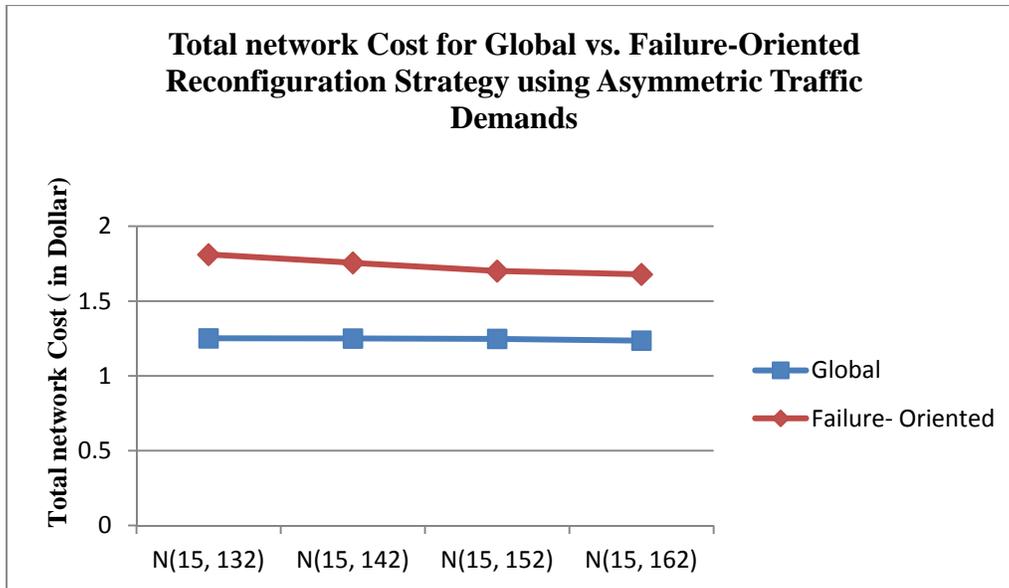


Figure 4.8: Total Network Cost for Global vs. Failure-Oriented Reconfiguration Strategy using asymmetric traffic demands

Figure 4.8 reveals that, in case of asymmetric traffic demands, the values of the total network cost in both reconfiguration strategies are also reflected by the values of capacity installation cost; consequently the total network cost decreases when the number of arcs increases.

In Figure 4.8, it can be observed that for asymmetric traffic demand, the global reconfiguration strategy performs better than the failure-oriented reconfiguration strategy. The reason behind this is that, in global reconfiguration, there are more options for the selection of candidate backup paths for the affected commodities than in the failure-oriented reconfiguration; consequently the capacity of the network can be shared in a better way.

4.2.1.5 CPU Time

With respect to CPU time, the results are captured in Table 4.10 and depicted in Figure 4.9.

Table 4.10: CPU Time (in Seconds) for Global vs. Failure-Oriented Reconfiguration Strategy using symmetric traffic demands

Networks	N(15, 132)	N(15, 142)	N(15, 152)	N(15, 162)
Global	0.22	0.22	0.26	0.36
Failure-Oriented	0.75	0.88	0.88	1.04

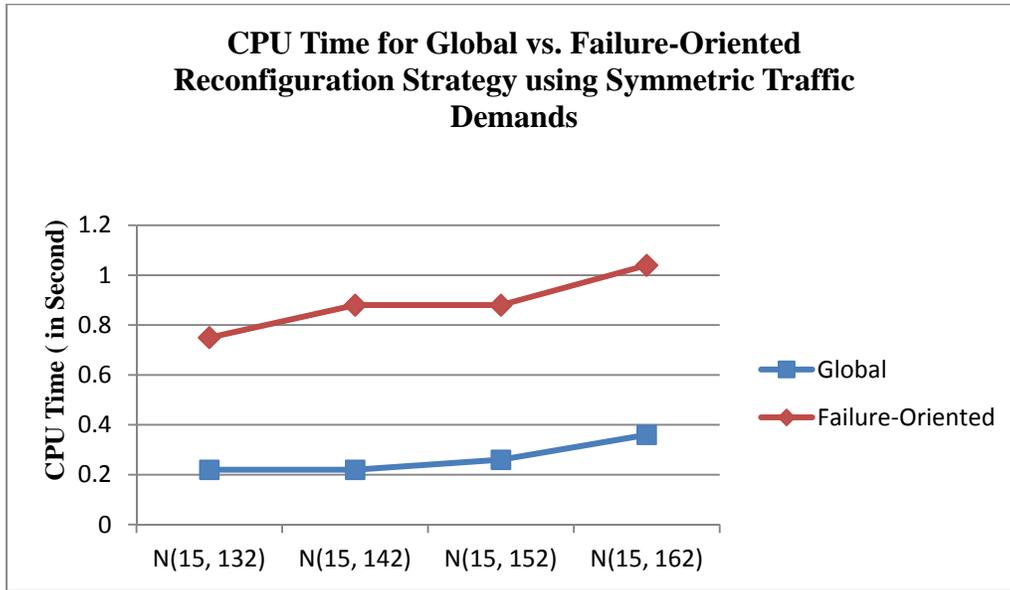


Figure 4.9: CPU Time for Global vs. Failure-Oriented Reconfiguration Strategy using symmetric traffic demands

Figure 4.9 shows that, our algorithms are quite fast as the global reconfiguration strategy can execute within only 0.36 seconds when restoring the affected traffics of a large network such as N(15,162) and in case of failure-oriented reconfiguration, it requires only 1.04 seconds.

The Table 4.11 summarizes the CPU Time in the case of asymmetric traffic demands.

Table 4.11: CPU Time (in Seconds) for Global vs. Failure-Oriented Reconfiguration Strategy using asymmetric traffic demands

Networks	N(15, 132)	N(15, 142)	N(15, 152)	N(15, 162)
Global	0.22	0.22	0.26	0.36
Failure-Oriented	0.74	0.88	0.88	1.04

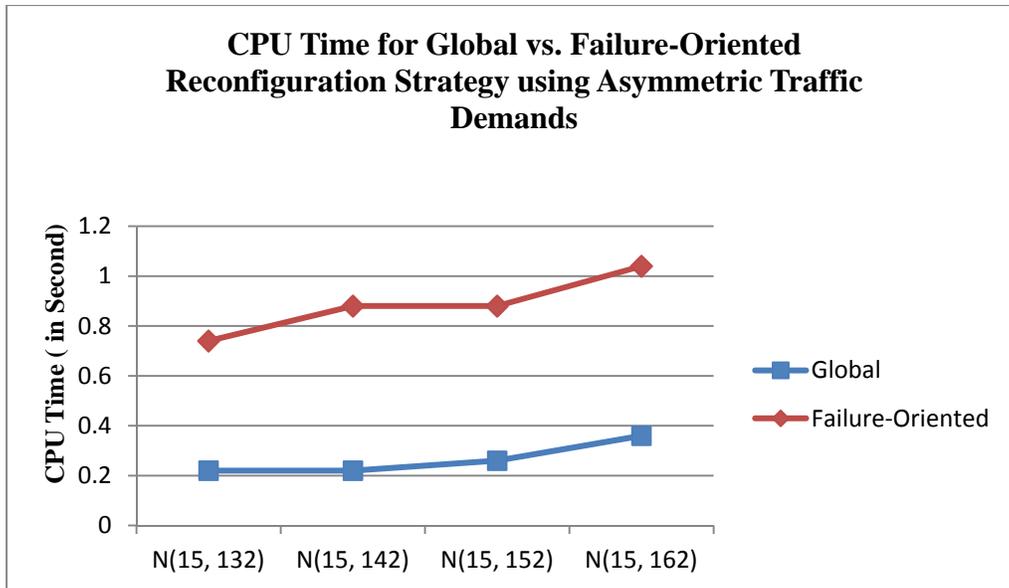


Figure 4.10: CPU Time for Global vs. Failure-Oriented Reconfiguration Strategy using asymmetric traffic demands

Similarly for asymmetric traffic demands, the global reconfiguration requires only 0.36 seconds for restoring the large network N (15,162) and the failure-oriented reconfiguration strategy needs only 1.04 seconds for restoring that same network.

4.2.2 Results Using Scenario 2

In scenario 2, the performance comparison between global reconfiguration strategy and failure-oriented reconfiguration strategy has been done using the following five networks: Network1, Network 2, Network 3, Network 4 and Network 5.

4.2.2.1 Capacity Installation Cost

The results for this metric using symmetric traffic demands are summarized in Table 4.12 and depicted in Figure 4.11.

Table 4.12: Capacity Installation Cost (in Dollars) for Global vs. Failure-Oriented Reconfiguration Strategy using symmetric traffic demands

Different Networks	Network 1	Network2	Network 3	Network 4	Network 5
Global	0.05	0.134	0.35	0.851	1.623
Failure-Oriented	0.13	0.339	0.879	2.1	3.989

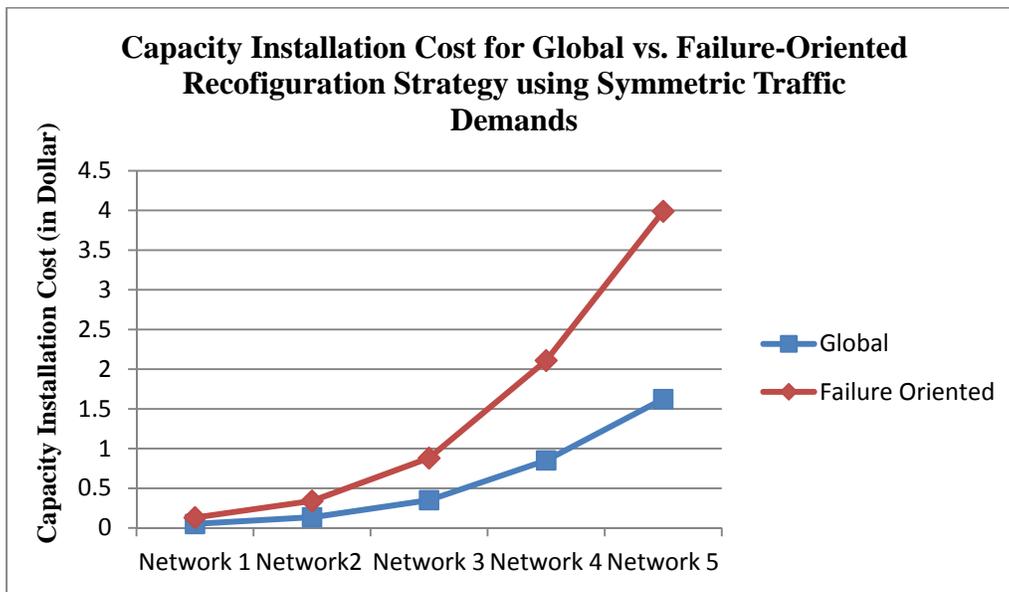


Figure 4.11: Capacity Installation Cost for Global vs. Failure-Oriented Reconfiguration Strategy using symmetric traffic demands

In Figure 4.11, it can be observed that, the capacity installation cost increases gradually when the network size increases for both restoration strategies. This can be justified by the fact that, when the network size is increased, the number of commodities and their traffic demands are also increased. To fulfill the increased traffic demands, required capacity and capacity installation cost are also increased.

It can also be seen in Figure 4.11, that the global reconfiguration requires less capacity installation cost than the failure-oriented reconfiguration. This can be attributed to the fact that, the global reconfiguration optimization procedure is more flexible for rerouting the affected traffics when a single link fails in the network than the failure-oriented reconfiguration optimization procedure, and the more flexible the optimization procedure is, the better the capacity is shared; consequently less capacity installation cost is likely to be required to optimize the network cost when using the global reconfiguration optimization procedure.

Another phenomenon revealed from the Figure 4.11 is that, the difference between global and failure-oriented reconfiguration strategies in terms of capacity installation cost increases with the increment in network size.

Table 4.13 captures the results of performance comparison between the two strategies with respect to capacity installation costs in the case of asymmetric traffic demands and Figure 4.12 depicts these results.

Table 4.13: Capacity Installation Cost (in Dollars) for Global vs. Failure-Oriented Reconfiguration Strategy using asymmetric traffic demands

Different networks	N (3,6)	N(5,20)	N(8,56)	N(12,126)	N(15,162)
Global	0.02	0.057	0.154	0.369	0.668
Failure-Oriented	0.052	0.1359	0.351	0.843	1.595

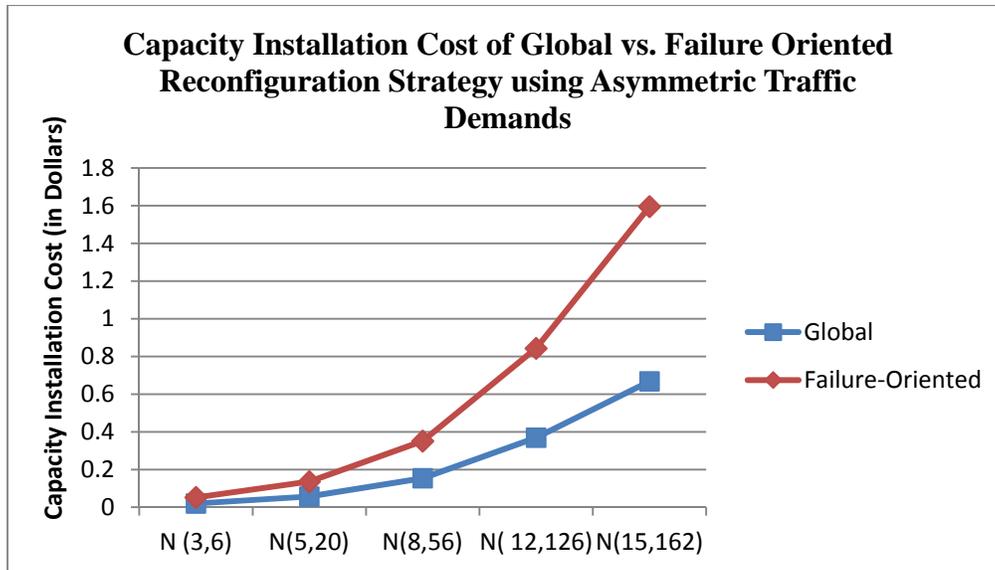


Figure 4.12: Capacity Installation Cost for Global vs. Failure-Oriented Reconfiguration Strategy using asymmetric traffic demands

In Figure 4.12, it can be observed that, global reconfiguration requires less capacity installation cost than failure-oriented reconfiguration strategy leading to a better sharing of the capacity. Again, this is attributed to the flexibility of the global reconfiguration optimization procedure in selecting candidate paths for the commodities. Also, the difference between global and failure-oriented reconfiguration is more pronounced for large size networks in terms of capacity installation cost.

4.2.2.2 Total Used Capacity

Table 4.14 summarizes the results of the performance metric total used capacity using symmetric traffic demands of the commodities, and Figure 4.13 depicts those results.

Table 4.14: Total Used Capacity (in Bits) for Global vs. Failure-Oriented Reconfiguration Strategy using symmetric traffic demands

Different networks	Network 1	Network2	Network 3	Network 4	Network 5
Global	50.124	134.162	350.26	851.671	1623.616
Failure-Oriented	130.006	339.987	879.939	2109.843	3989.771

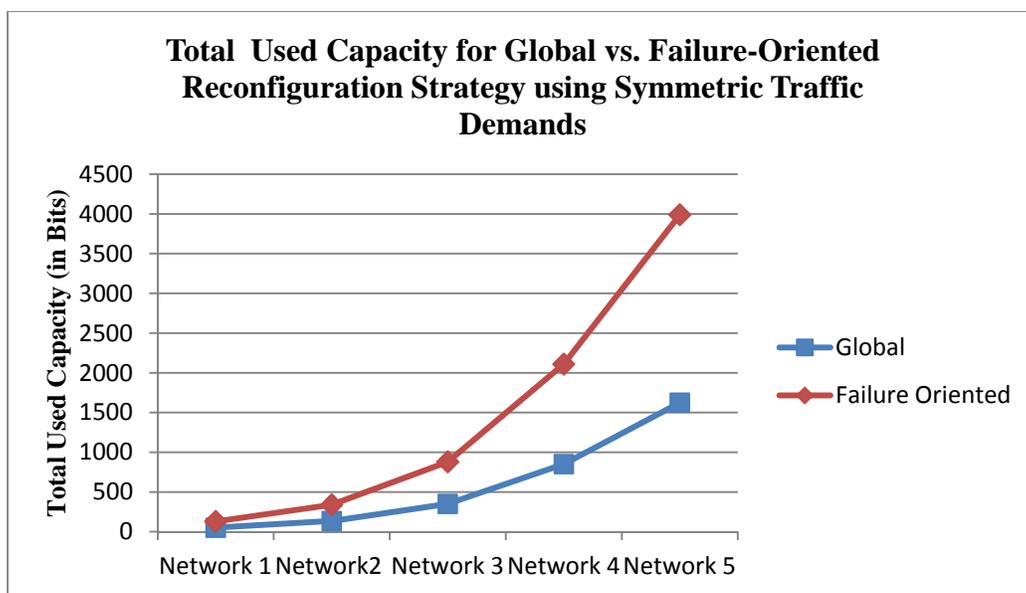


Figure 4.13: Total Used Capacity for Global vs. Failure-Oriented Reconfiguration Strategy using symmetric traffic demands

Figure 4.13 reveals the expected results for both reconfiguration strategies because with the increase in network size, the total required capacity to fulfill the increased traffic demands and optimize the network cost is also increased.

Another phenomenon revealed from the Figure 4.13 is that, global reconfiguration requires less capacity than failure-oriented reconfiguration strategy. This can be justified by the fact that, in global reconfiguration strategy, the whole layout of the working paths can be

rearranged in order to recover from the single link failure whereas in failure-oriented reconfiguration, only the affected traffics are restored. In this sense, global reconfiguration can better share the capacity among the arcs. Moreover, it can also be seen from Figure 4.13 that, the difference between the strategies increases when the network size increases.

Table 4.15 summarizes the results of the performance comparison in terms of total capacity for asymmetric traffic demands.

Table 4.15: Total Used Capacity (in Bits) for Global vs. Failure-Oriented Reconfiguration Strategy using asymmetric traffic demands

Different networks	N (3,6)	N(5,20)	N(8,56)	N(12,126)	N(15,162)
Global	20	57	154	369	668
Failure-Oriented	52	135.9	351	843	1595

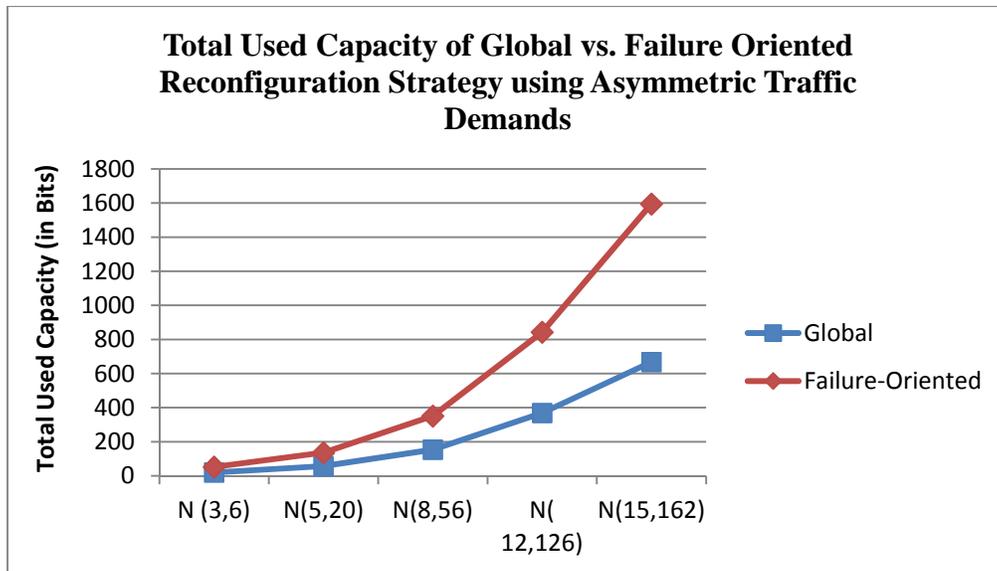


Figure 4.14: Total Used Capacity for Global vs. Failure-Oriented Reconfiguration Strategy using asymmetric traffic demands

In Figure 4.14, it can be observed that, for asymmetric traffic demands, the same trend as in the case of symmetric traffic demands prevails.

4.2.2.3 Routing Cost

Table 4.16 summarizes the results obtained for the routing cost performance metric when using symmetric traffic demands. These results are depicted in Figure 4.15.

Table 4.16: Routing Cost (in Dollars) for Global vs. Failure-Oriented Reconfiguration Strategy using symmetric traffic demands

Different networks	Network 1	Network2	Network 3	Network 4	Network 5
Global	0.0158	0.085	0.265	0.602	0.783
Failure-Oriented	0.003	0.01	0.028	0.063	0.082

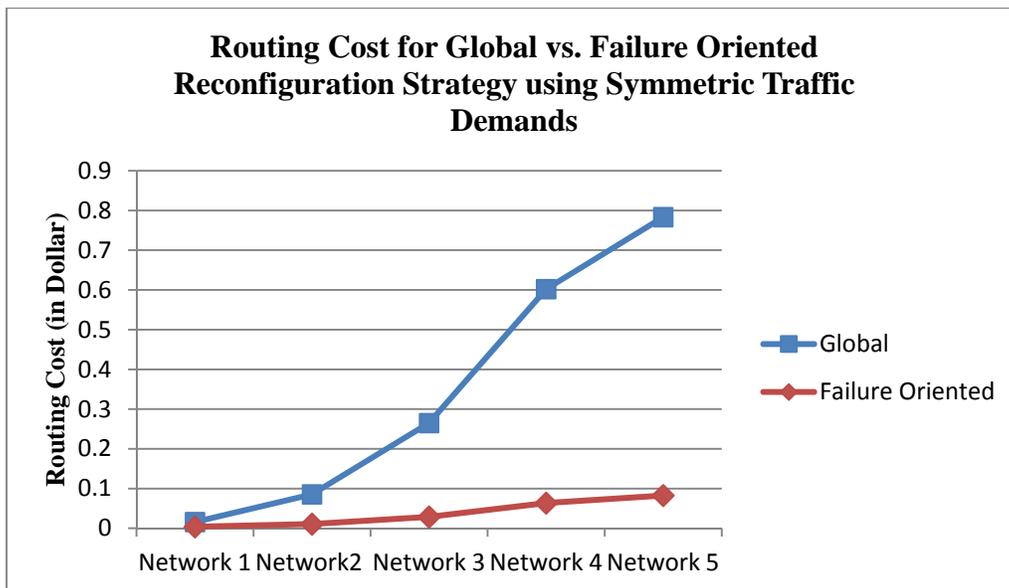


Figure 4.15: Routing Cost for Global vs. Failure-Oriented Reconfiguration Strategy using symmetric traffic demands

Figure 4.15 reveals that the routing cost increases when the network size increases for both reconfiguration strategies. This might be due to the fact that, as the network size increases in respect of commodities and number of arcs, the traffic demands are also increased; consequently the cost for routing these increased traffic demands are also increased.

In Figure 4.15, it can also be observed that the global reconfiguration requires higher routing cost than failure-oriented reconfiguration. This is due to the fact that, in global reconfiguration, all the traffics are rerouted for recovering from the failure where as in the failure-oriented reconfiguration strategy; only the affected traffic flows are rerouted.

Also in Figure 4.15, the difference between global and failure-oriented reconfiguration increases gradually with the increment in network size. For small networks, the difference of routing cost between the two strategies is quite small.

In case of asymmetric traffic demands, the same trend for routing cost as in the case of symmetric traffic demands is observed. The results are summarized in Table 4.17 and depicted in Figure 4.16.

Table 4.17: Routing Cost (in Dollars) for Global vs. Failure-Oriented Reconfiguration Strategy using asymmetric traffic demands

Different networks	N (3,6)	N(5,20)	N(8,56)	N(12,126)	N(15,162)
Global	0.035	0.113	0.315	0.751	1.23
Failure-Oriented	0.055	0.146	0.38	0.907	1.678

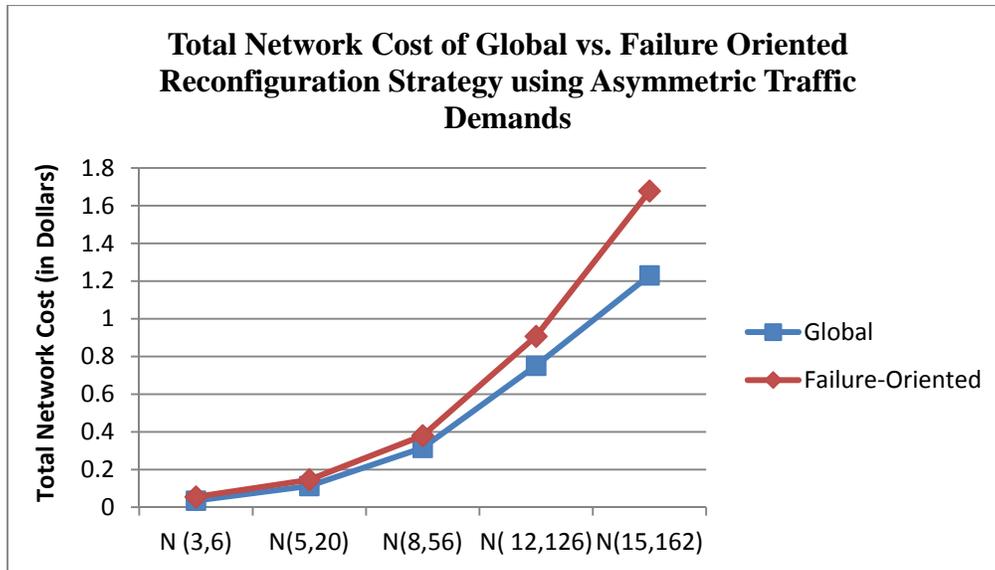


Figure 4.16: Routing Cost of Global vs. Failure-Oriented Reconfiguration Strategy using asymmetric traffic demands

4.2.2.4 Total Network Cost

The results obtained for comparing the global and failure-oriented reconfiguration strategies in terms of total network cost are captured in Table 4.18 and depicted in Figure 4.17 when symmetric traffic demands are used.

Table 4.18: Total Network Cost (in Dollars) for Global vs. Failure-Oriented Reconfiguration Strategy using symmetric traffic demands

Different networks	Network 1	Network2	Network 3	Network 4	Network 5
Global	0.065	0.219	0.615	1.453	2.4
Failure-Oriented	0.133	0.35	0.908	2.173	4.072

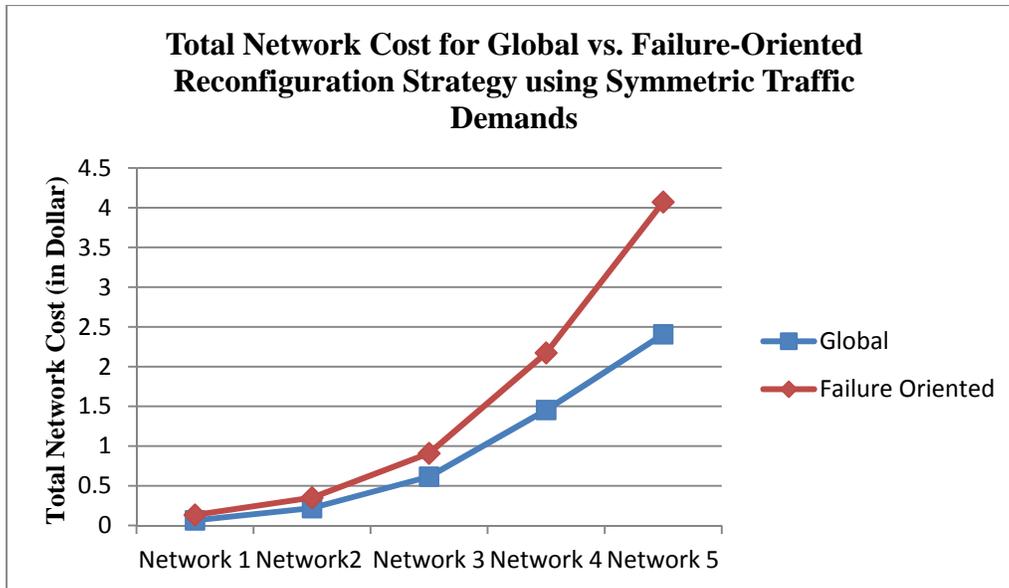


Figure 4.17: Total Network Cost for Global vs. Failure-Oriented Reconfiguration Strategy using symmetric traffic demands

In Figure 4.17, it can be observed that, when the network size increases, the total network cost also increases for both reconfiguration strategies. This might be due to the fact that, when network size increase in terms of commodities and number of arcs, network cost also increases to maintain the increased traffic requirements.

Figure 4.17 also shows that, the global reconfiguration generates less network cost than the failure-oriented reconfiguration strategy. This can be attributed to the fact that, as the global reconfiguration has a more flexible optimization procedure than the failure-oriented reconfiguration, it can better share the capacity among the arcs; consequently generates less network cost than the failure-oriented reconfiguration strategy.

With an increment in the network size, the total network cost of both strategies is also increased in the case of asymmetric traffic demands. The results are summarized in Table 4.19 and depicted in Figure 4.18.

Table 4.19: Total Network Cost (in Dollars) for Global vs. Failure-Oriented Reconfiguration Strategy using asymmetric traffic demands

Different networks	Network 1	Network2	Network 3	Network 4	Network 5
Global	0.008	0.025	0.052	0.117	0.207
Failure-Oriented	0.009	0.027	0.073	0.167	0.267

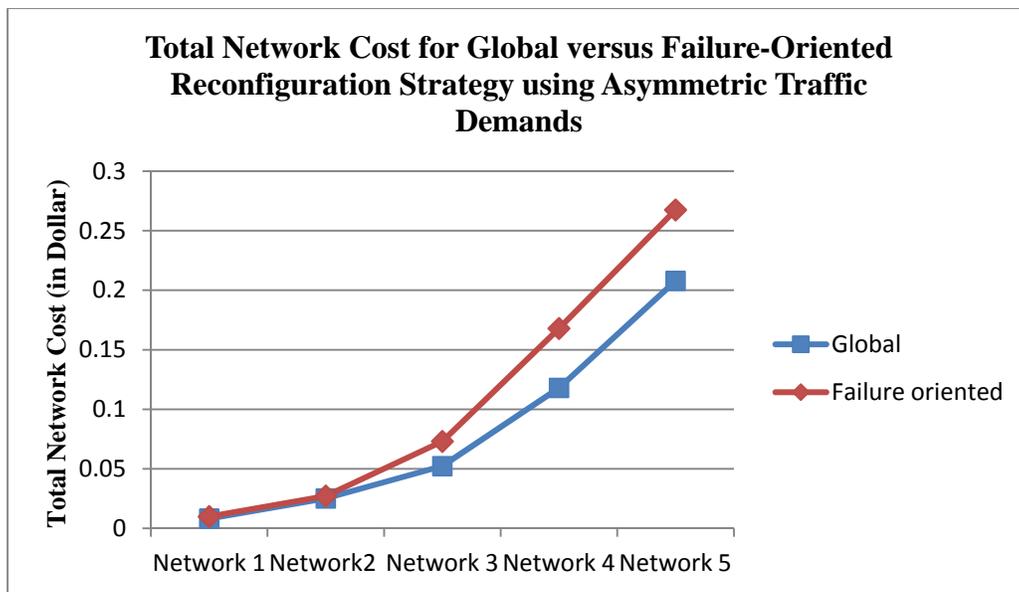


Figure 4.18: Total Network Cost for Global vs. Failure-Oriented Reconfiguration Strategy using asymmetric traffic demands

In Figure 4.18, it is observed that, as expected, global reconfiguration generates less total network cost than failure-oriented reconfiguration strategy and the difference between global and failure-oriented reconfiguration in terms of total network cost becomes quite significant for large size networks.

4.2.2.5 CPU Time

The results obtained from the performance metric CPU time are captured in Table 4.20 and depicted in Figure 4.19 when symmetric traffic demands are considered.

Table 4.20: CPU Time (in Seconds) for Global vs. Failure-Oriented Reconfiguration Strategy using symmetric traffic demands

Different networks	Network 1	Network2	Network 3	Network 4	Network 5
Global	0.001	0.01	0.02	0.11	0.36
Failure-Oriented	0.02	0.04	0.17	0.64	1.04

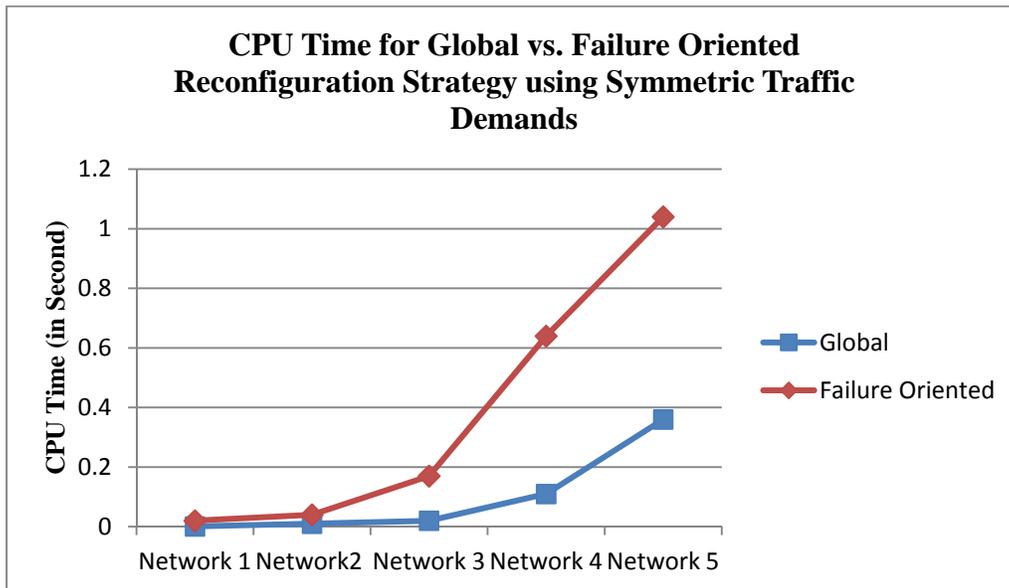


Figure 4.19: CPU Time for Global vs. Failure-Oriented Reconfiguration Strategy using symmetric traffic demands

Figure 4.19 reveals that both reconfiguration strategies are quite fast as the convergence time for the global reconfiguration strategy is only 0.36 seconds for restoring the affected traffics

of the large network i.e. Network 5 and for the failure-oriented reconfiguration strategy, the convergence time is 1.04 seconds only.

In case of asymmetric traffic demands, the CPU time is captured in Table 4.21.

Table 4.21: CPU Time (in Seconds) for Global vs. Failure-Oriented Reconfiguration Strategy using asymmetric traffic demands

Different networks	Network 1	Network2	Network 3	Network 4	Network 5
Global	0.001	0.005	0.02	0.11	0.19
Failure-Oriented	0.01	0.02	0.1	0.38	1.12

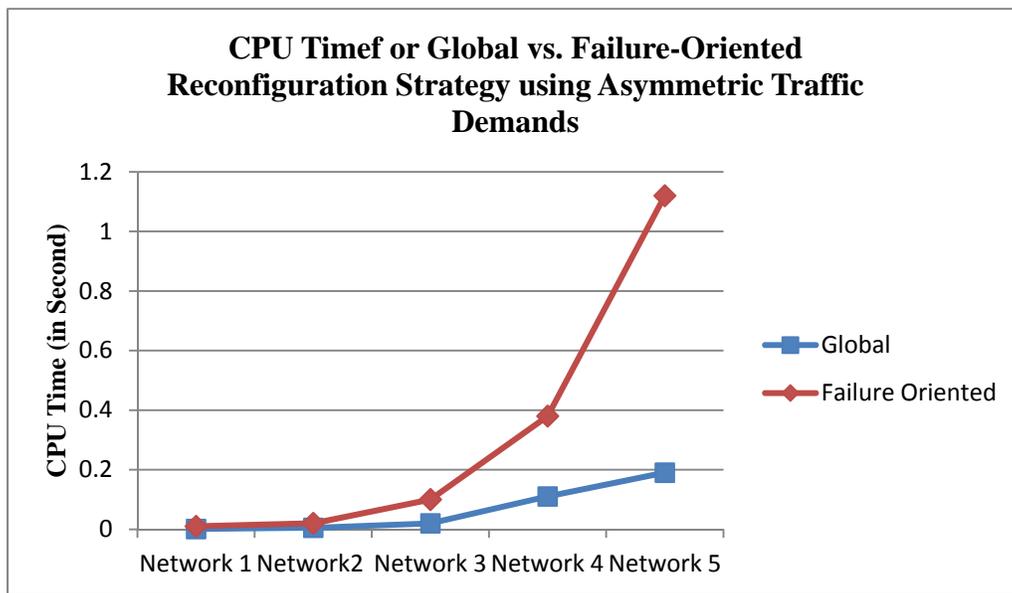


Figure 4.20: CPU Time for Global vs. Failure-Oriented Reconfiguration Strategy using asymmetric traffic demands

Figure 4.20 shows that, the global reconfiguration strategy executes within 0.36 seconds whereas the failure-oriented reconfiguration strategy takes 1.04 seconds for the large network i.e. Network 5 for asymmetric traffic demands.

4.2.3 Results Using Scenario 3

In this section, the performance comparison between the global and failure-oriented reconfiguration strategies has been analyzed in terms of the same five performance metrics. In this scenario, various traffic demands of the commodities are considered for a fixed large network i.e. N(15,152).

4.2.3.1 Capacity Installation Cost

For analyzing the performance comparison between the global and failure-oriented reconfiguration strategies with respect to capacity installation cost, the obtained results are summarized in Table 4.22 and depicted in Figure 4.21.

Table 4.22: Capacity Installation Cost (in Dollars) for Global vs. Failure-Oriented Reconfiguration Strategy in different traffic demands

Traffic Demands	0.1	0.2	1	2	5	10
Global	0.038	0.076	0.358	0.696	1.7	3.206
Failure-Oriented	0.171	0.212	0.811	1.623	4.059	8.119

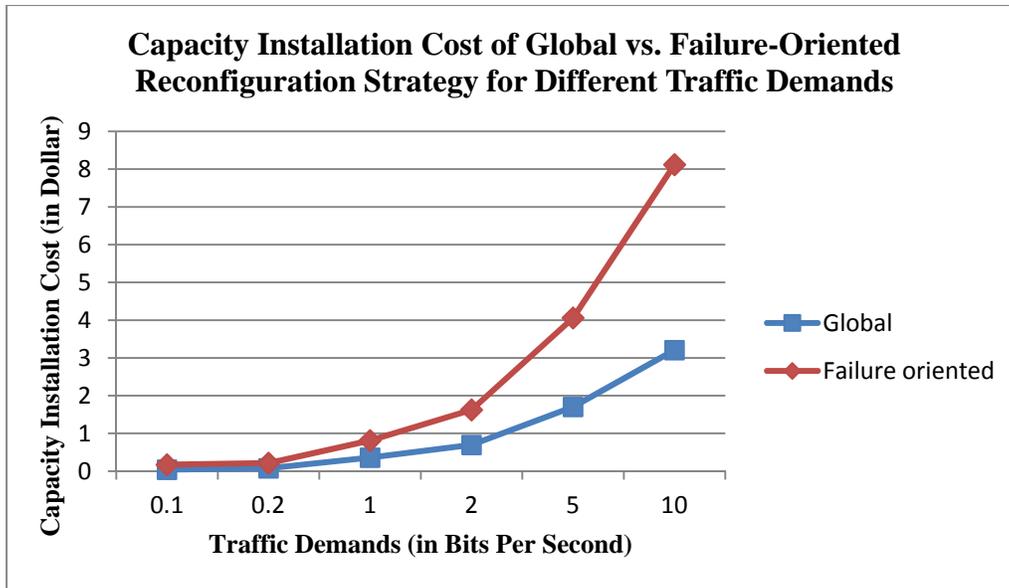


Figure 4.21: Capacity Installation Cost for Global vs. Failure-Oriented Reconfiguration Strategy in different traffic demands

It can be observed from Figure 4.21 that the global reconfiguration needs less capacity installation cost than the failure-oriented reconfiguration strategy for various bandwidth requirements of the commodities for the same network. The reason behind this observation is that, the global reconfiguration optimization procedure is more flexible in selecting reconfiguration routes and better shares the capacity than the failure-oriented; as a result, global needs less capacity installation cost than failure-oriented reconfiguration. Moreover, it can be seen that, the difference between the global and failure-oriented reconfiguration strategy becomes larger as the bandwidth requirements increases.

4.2.3.2 Total Used Capacity

Table 4.23 summarizes the obtained results for comparing the strategies in terms of required capacity for different traffic requirements.

Table 4.23: Total Used Capacity (in Bits) for Global vs. Failure-Oriented Reconfiguration Strategy in different traffic demands

Traffic Demands	0.1	0.2	1	2	5	10
Global	38.932	76.279	358.318	696.066	1702.8	3206.83
Failure-Oriented	171.378	212.756	811.871	1623.88	4059.81	8119.77

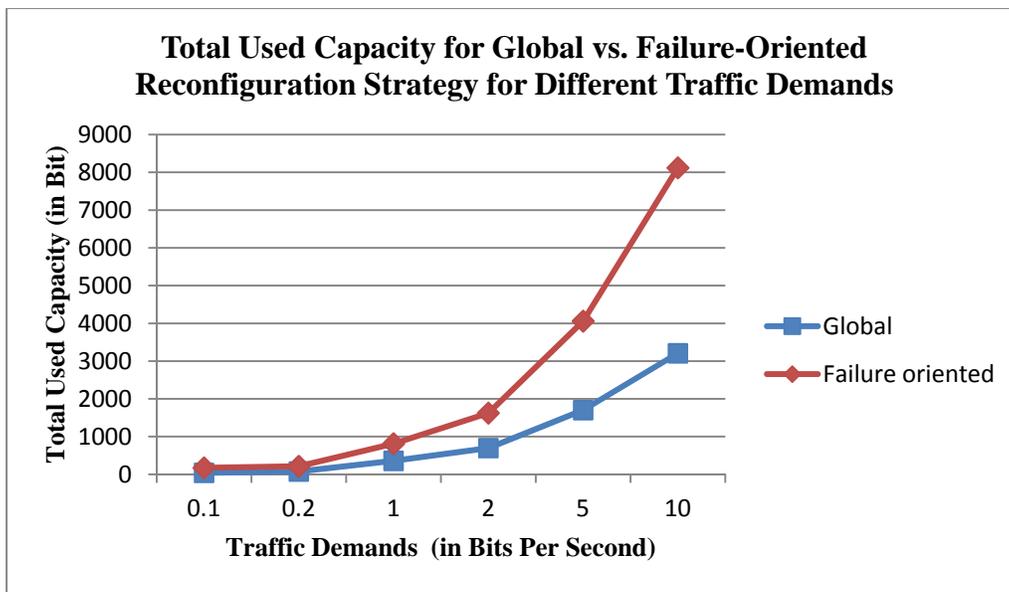


Figure 4.22: Total Used Capacity for Global vs. Failure-Oriented Reconfiguration Strategy in different traffic demands

The observations of Figure 4.22 can be understandable, as the more flexible the optimization procedure is, the less capacity is likely to be required for network optimization. As the global is more flexible than the failure-oriented optimization procedure, global requires less capacity than the failure-oriented reconfiguration for any bandwidth requirement. Also, Figure 4.22 shows that, the difference between the global and failure-oriented reconfiguration strategy increases significantly for large traffic demands.

4.2.3.3 Routing Cost

The results for performance analysis of the restoration strategies in terms of routing cost are captured in Table 4.24 and depicted in Figure 4.23.

Table 4.24: Routing Cost (in Dollars) for Global vs. Failure-Oriented Reconfiguration Strategy in different traffic demands

Traffic Demands	0.1	0.2	1	2	5	10
Global	0.11	0.16	0.4537	0.55117	0.74926	0.93117
Failure-Oriented	0.02791	0.04518	0.07669	0.079	0.081	0.083

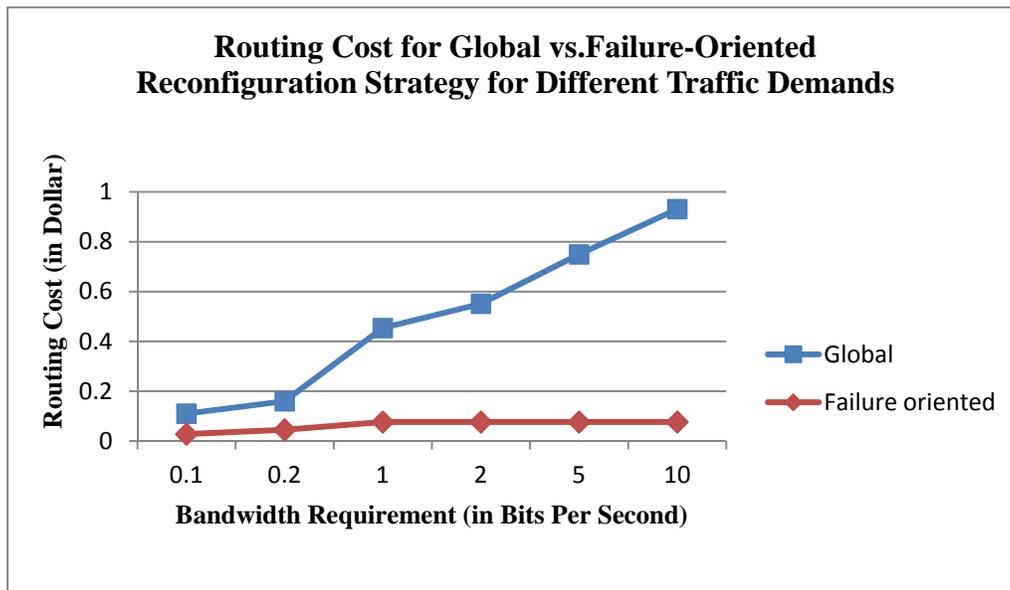


Figure 4.23: Routing Cost for Global vs. Failure-Oriented Reconfiguration Strategy in different traffic demands

It can be observed from Figure 4.23 that, the global reconfiguration strategy requires more routing cost than the failure-oriented reconfiguration due to the fact that the global reconfiguration reroutes all traffics for failure recovery whereas the failure-oriented

reconfiguration reroutes affected traffics only. One more phenomenon revealed in Figure 4.23 is that the difference between the restoration strategies in terms of routing cost becomes worse with the increment of bandwidth requirements.

4.2.3.4 Total Network Cost

The results obtained for comparing the global and failure-oriented reconfiguration strategies with respect to total network cost are summarized in Table 4.25.

Table 4.25: Total Network Cost (in Dollars) for Global vs. Failure-Oriented Reconfiguration Strategy in different traffic demands

Traffic Demands	0.1	0.2	1	2	5	10
Global	0.14893	0.23628	0.81202	1.24723	2.45206	4.138
Failure oriented	0.19929	0.25794	0.88857	1.70056	4.13649	8.19645

Table 4.25 summarizes that, the total network cost gradually increases when the traffic requirements increase gradually for both restoration strategies. This can be justified by the fact that, when the traffic demands are increased in a network, the network cost to fulfill the traffic demands should also increase. Table 4.25 also shows that, the difference between the two restoration strategies is more pronounced in terms of total network cost for a large bandwidth requirement of the commodities.

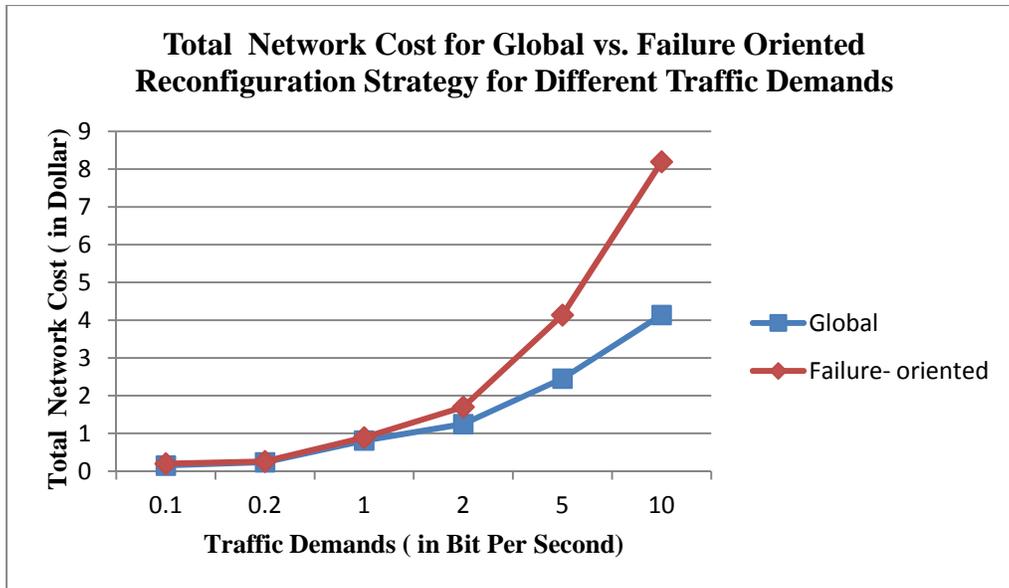


Figure 4.24: Total Network Cost for Global vs. Failure-Oriented Reconfiguration Strategy in different traffic demands

It can also be observed from Figure 2.4 that, in terms of network cost, global reconfiguration is preferable to failure-oriented reconfiguration for any bandwidth requirement. It might be due to the fact that, as the global reconfiguration optimization procedure is more flexible in selecting candidate paths, it requires less network cost to optimize the network than the failure-oriented reconfiguration strategy.

4.2.3.5 CPU Time

The performance metric CPU time for both restoration strategies are summarized in Table 4.26.

Table 4.26: CPU Time (in Seconds) for Global vs. Failure-Oriented Reconfiguration Strategy in different traffic demands

Traffic Demands	0.1	0.2	1	2	5	10
Global	0.08	0.12	0.2	0.26	0.26	0.26
Failure-Oriented	0.88	0.88	0.88	0.88	0.88	0.88

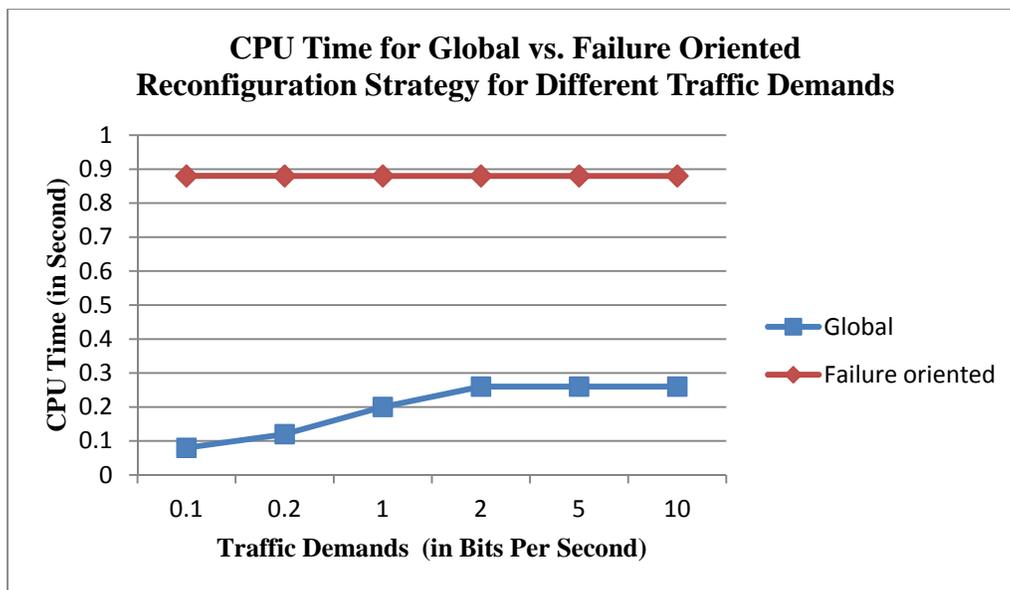


Figure 4.25: CPU Time for Global vs. Failure-Oriented Reconfiguration Strategy in different traffic demands

Figure 4.25 depicts that both restoration strategies can converge very fast for a large network $N(15,152)$ for various traffic requirements. Within 0.26 seconds, the global reconfiguration strategy can be executed with a traffic requirement of 10 bits per second. For the failure-oriented reconfiguration strategy, it also takes only 0.88 seconds to converge for the same bandwidth requirement.

Chapter 5

Conclusion

This thesis has studied the problem of CFA in self-healing ATM network using path restoration strategy. Two nonlinear mathematical models are presented, one for the global reconfiguration strategy and the other for the failure-oriented reconfiguration strategy. These restoration strategies are compared quantitatively in terms of five performance metrics which are: capacity installation cost, routing cost, total cost of the network which is in fact summation of capacity installation cost and routing cost, total used capacity and the required CPU time for convergence. For the performance analysis, different network scenarios, different network topologies, both symmetric and asymmetric traffic demands are considered.

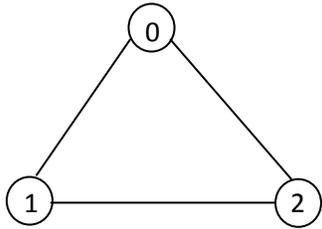
The findings from the numerical results of the thesis can be summarized as follows:

- The global reconfiguration strategy always performs better than the failure-oriented reconfiguration strategy in terms of total network cost, total required capacity and CPU time.
- The difference between the global and the failure-oriented reconfiguration strategies is less pronounced in networks having large number of arcs in terms of required capacity and capacity installation cost. But in terms of routing cost, the difference between these two strategies is more evident in a network having large number of arcs.

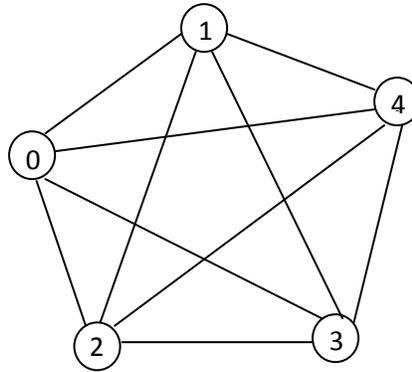
- For a small bandwidth requirement, the difference between the global and failure-oriented reconfiguration strategies is quite small whereas the difference is more pronounced for a large bandwidth requirement.
- With the gradual increment of the number of arcs of a network, the capacity installation cost, total required capacity and the total network cost are gradually decreased for both the restoration strategies using symmetric and asymmetric traffic demands.
- When the bandwidth requirements of the commodities are increased, all the five performance metrics of the two restoration strategies are also increased.

In future, it is desirable to consider the proposed mathematical models and their solution approaches in the context of the generalized multi-protocol label switching (GMPLS) networks. It will also be interesting to consider the node failure event when dealing with survivable network designs.

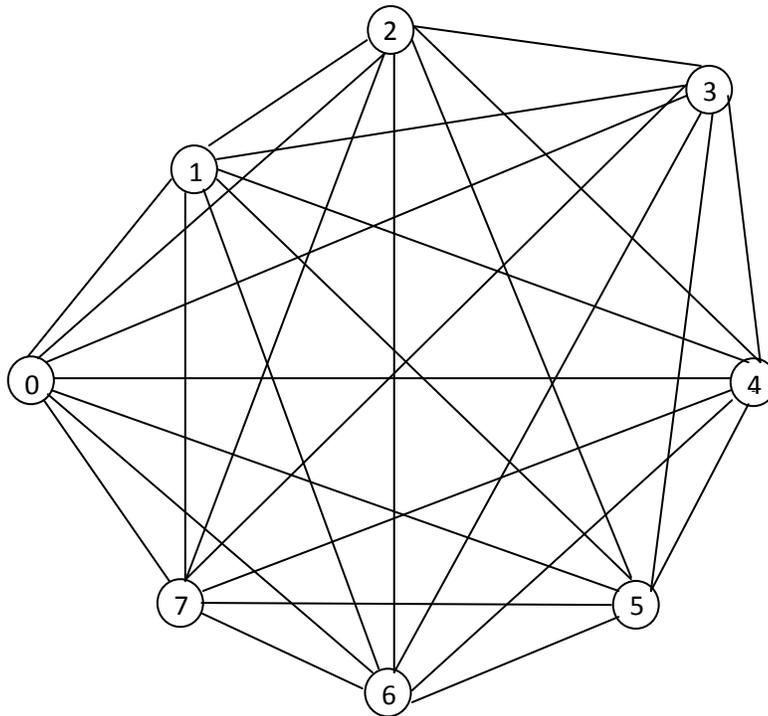
Appendix A



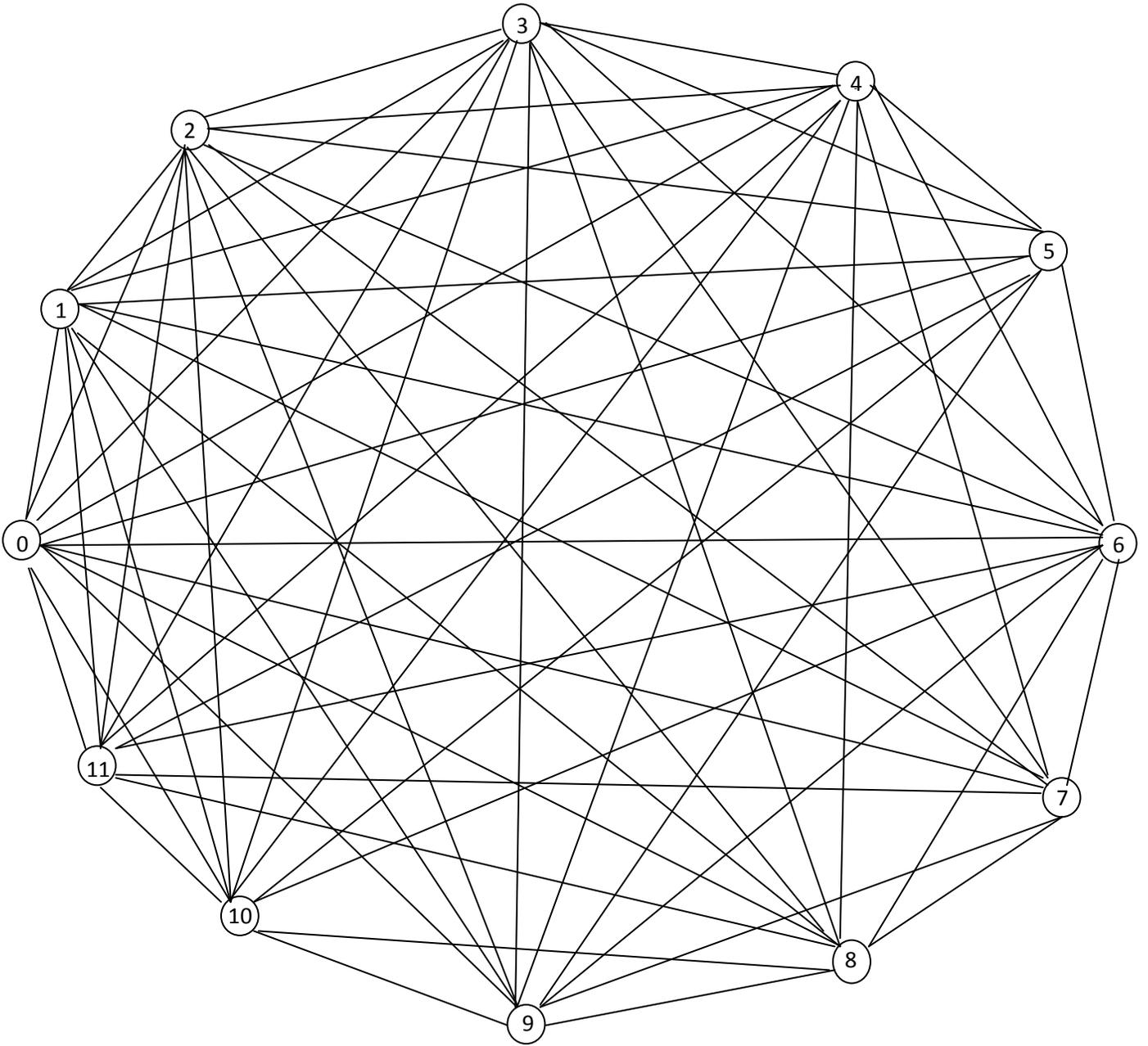
Network 1 with 3 nodes and 6 arcs



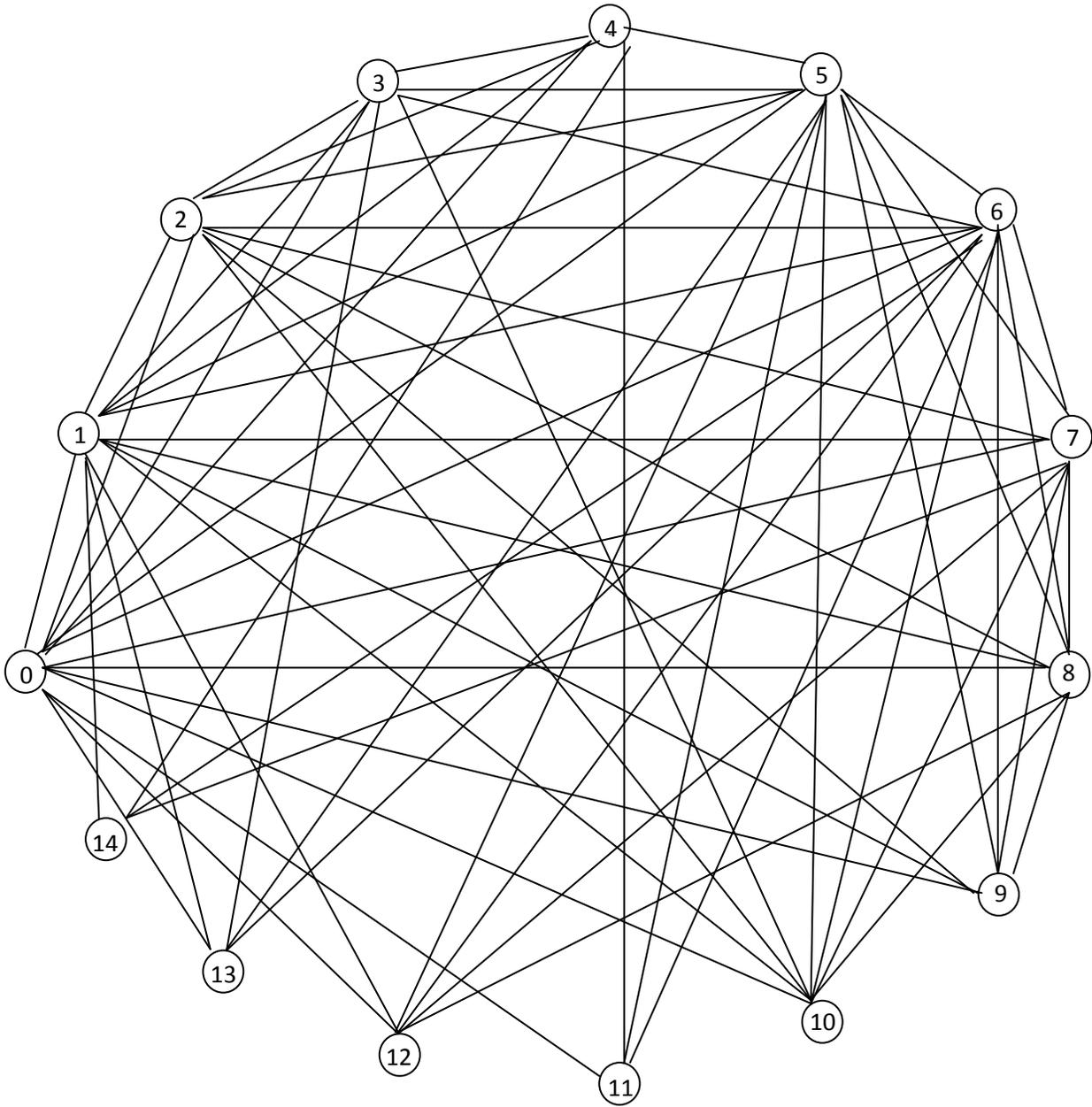
Network 2 with 5 nodes and 20 arcs



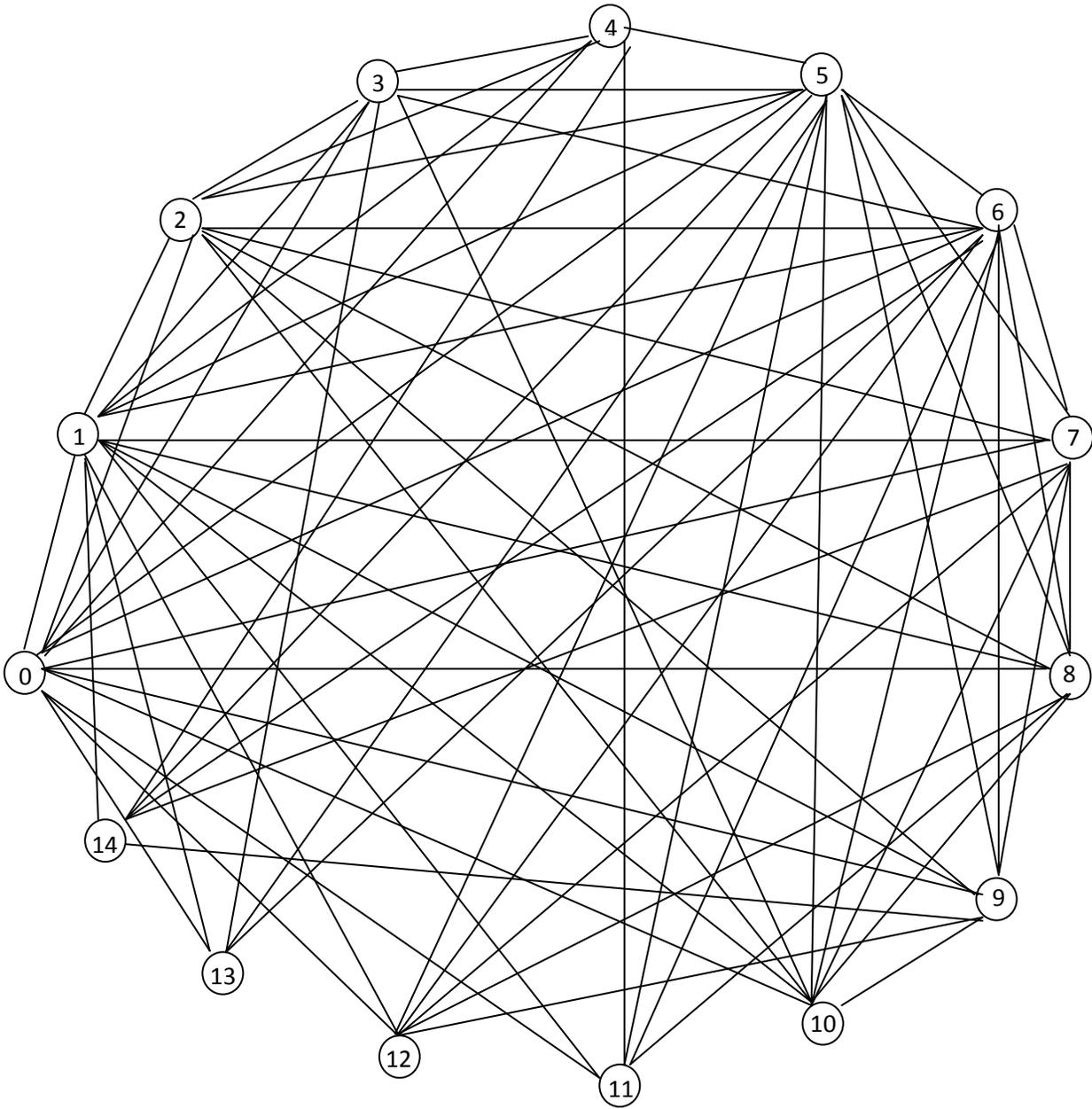
Network 3 with 8 nodes and 56 arcs



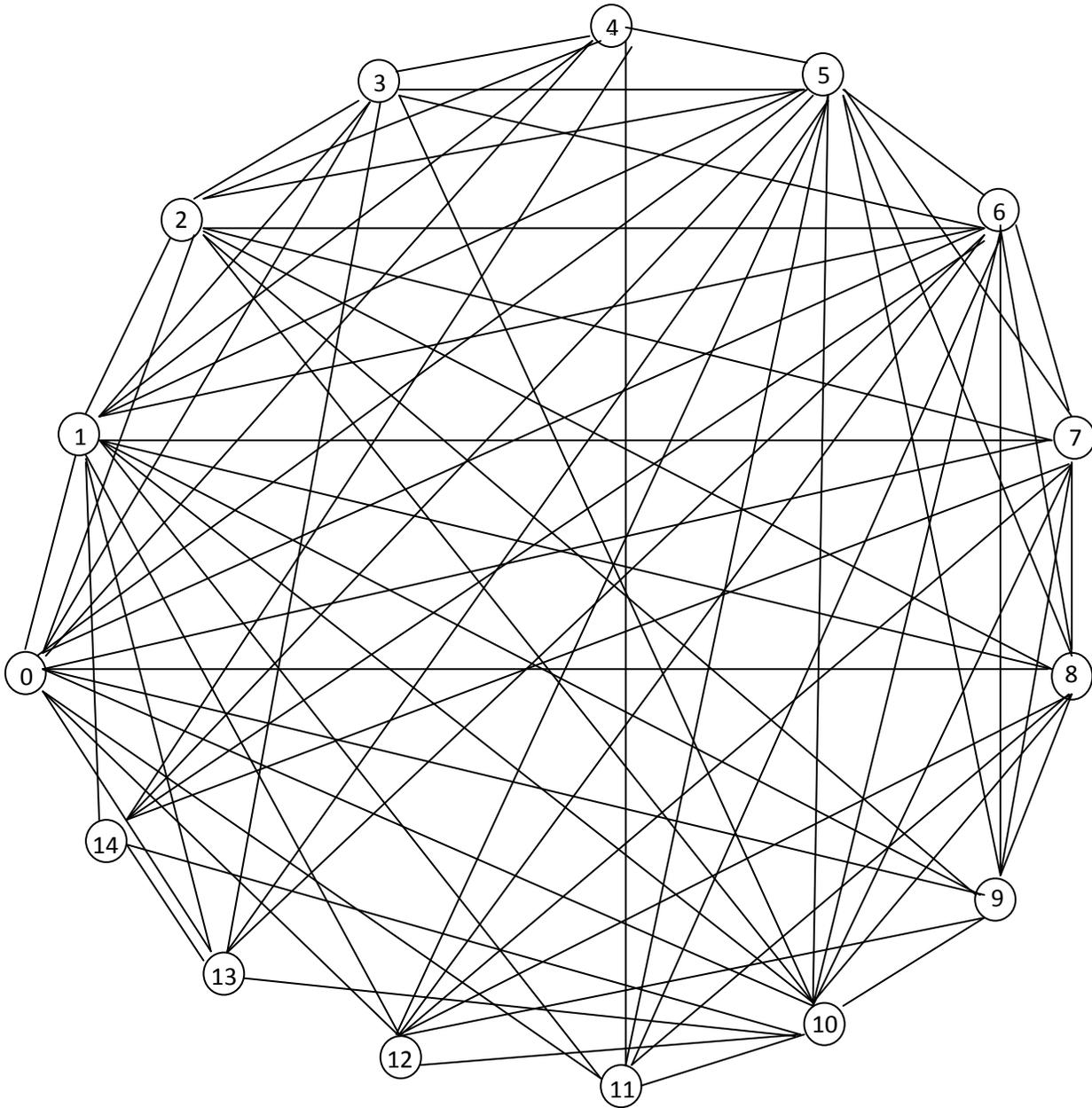
Network 4 with 12 nodes and 126 arcs



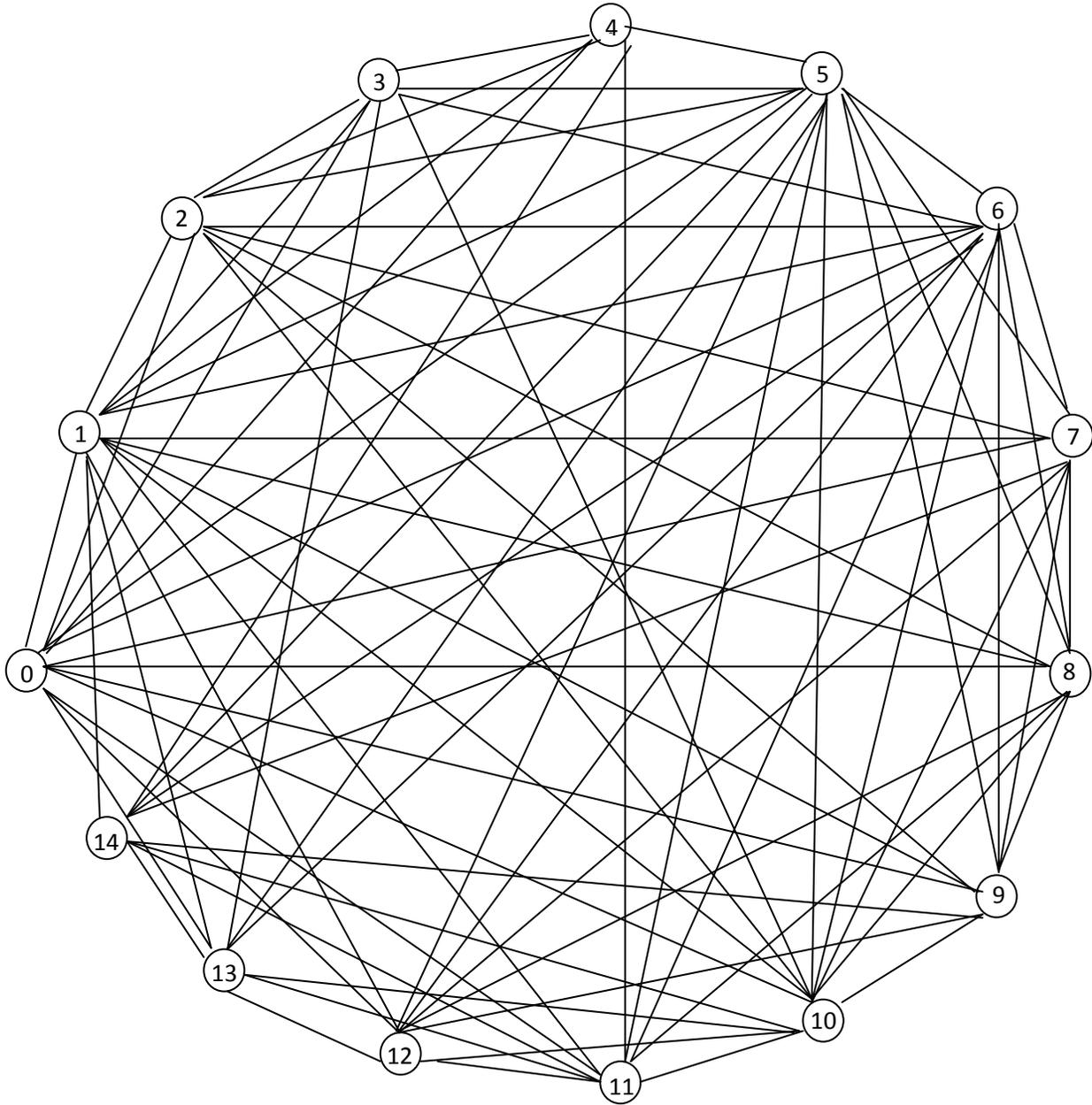
N (15,132), Network with 15 nodes and 132 arcs



N (15,142), Network with 15 nodes and 142 arcs



N (15,152), Network with 15 nodes and 152 arcs



N (15,162) or Network 5 with 15 nodes and 162 arcs

References

- [1] S. Kwong, D.W.F. Lam, K.S. Tang and K-F. Man, "Optimization of Spare Capacity in Self-Healing Multicast ATM Network Using Genetic Algorithm", *IEEE Transactions on Industrial Electronics*, vol. 47, No. 6, pp.1334-1343, Dec. 2000.
- [2] K. Murakami and S. Hyong, "Comparative Study on Restoration Schemes of Survivable ATM Networks", in *Proceedings of International Conference on Computer Communications*, vol.1, pp.345-352, Kobe, Japan, April 1997.
- [3] Y. Xiong and L.G. Masson, "Restoration Strategies and Spare Capacity Requirements in Self-Healing ATM Networks", *IEEE/ACM Transactions on Networking*, vol. 7, no. 1, pp. 98-110, Feb. 1999.
- [4] Y. Liu, D. Tipper and P. Siripongwutikorn, "Approximating Optimal Spare Capacity Allocation by Successive Survivable Routing", *IEEE/ACM Transaction on Networking*, vol. 13, issue 1, pp. 198-211, Feb. 2005.
- [5] K. Murakami and S. Hyong, "Joint Optimization of Capacity and Flow Assignment for Self-Healing ATM Networks", in *Proceedings of IEEE International Communications Conference*, pp. 216-220, Seattle, WA, 1995.
- [6] D. Rajan and A. Atamturk, "Survivable Network Design: Routing of Flows and Slack" in *Telecommunications Network Design and management, Sixth INFORMS Telecommunications Conference*, pp. 65-81, Florida, USA, March 2003.
- [7] B. A. Forouzan, *Data Communications and Networking*. 4thed. New York: McGraw-Hill, 2007, pp. 523-535, ISBN 978-0-07-296775-3.

- [8] I. Woungang, S. Misra, and M. S. Obaidat, "On the Problem of Capacity Allocation and Flow Assignment in Self-healing ATM networks", *Journal of Computer Communications*, vol. 30, Issue 16, pp. 3169-3178, 2007.
- [9] L. Nederlof, K. Struyue, H. Misser Chris Shea, Y. Du, and B. Tamayo, "End-to-end Survivable Broadband networks," *IEEE Communications Magazine*, pp. 63-70, 1995.
- [10] W. -P. Wang, D. Tipper, B. Jxger, and D. Medhi, " Faulty Recovery Routing in Wide Area Packet networks," in *Proceedings of 15th International Teletraffic Congress*, Washington, DC, June 1997.
- [11] B. Jxger and D. Tipper, "On Fault Recovery Priority in ATM networks," in *Proceedings of IEEE International Communications Conference(ICC'98)*, vol.3, pp. 1410-1414, Atlanta, GA, June 1998.
- [12] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, Mass.: Athena Scientific, 1999, ISBN 1-886529-00-0.
- [13] T-H.Wu, *Fiber Network Service Survivability*. Norwood, MA: Artech House, 1992, ch.4.
- [14] Isaac Woungang, Guangyan Ma, Mieso K. Denko, Sudip Misra, Han-Chieh Chao and Mohammad S. Obaidat, "Survivable ATM Mesh networks: Techniques and performance Evaluation", *Journal of Systems and Software*, vol. 83, Issue 3, pp. 457-466, 2010.
- [15] C. Y. Lee and S.J. Koh, "A Design of Self-Healing ATM Networks Based on Backup Virtual Paths", *Computers and Operations Research*, vol. 25, Issues 7-8, pp. 595-609, July 1998.
- [16] C-S Wu, S-W Lee, Y-T Hou and Y-S Chu, "A New Preplanned Self-healing Scheme for Multicast ATM Network", in *Proceeding of International Conference in Communication and Technology*, pp. 888-891, Beijing, May 1996.

- [17] R. Kawamura, K. Sato and I. Tokizawa, "Self-Healing ATM Networks Based on Virtual Path Concept", *IEEE J. Select. Areas Commun.*, vol. 12, no.1, pp. 120-127, Jan. 1994.
- [18] K. Murakami and S. Hyong, "Optimal Capacity and Flow Assignment for Self-Healing ATM Networks Based on Line and End-to-End Restoration", *IEEE/ACM Transaction on Networking*, vol. 6, no. 2, pp. 207-221, April 1998.
- [19] K. Murakami and H. S .Kim, "Virtual Path Routing for Survivable ATM Networks," *IEEE/ACM Trans. on Networking*, vol.4, no.1, pp 22-39, Jan. 1996.
- [20] L. Kleinrock, *Queuing systems*, vol. 1, New York: Wiley Interscience, 1975, ISBN 978-0471491101.
- [21] A. Hamdi, P. Mahey and J.-P. Dussault, " A New Decomposition Method in Non convex Programming", *Recent Advances in Optimization*, P. Gritzmann, R. Horst, E. Sachs and R. Tichatschke eds., Lecture Notes in Economics and Mathematical Systems, vol. 452, pp. 90-104, 1997.
- [22] P. E. Frandsen, K. Jonasson, H.B. Nielsen, and O. Tingleff, *Unconstrained Optimization, Informatics and Mathematical Modelling*, 3rd ed. Lyngby, Denmark: Technical University of Denmark, 2004, ch.5, sec.5.1, pp.45-46.
- [23] C. T. Kelly, *Iterative Methods for Optimization*, in: *Frontiers in Applied Mathematics*, vol.18, Philadelphia, PA: SIAM, 1999, ISBN 978-0898714333.
- [24] J. Sosnosky, "Service applications for SONET DCS distributed restoration," *IEEE J. Select. Areas Commun.*, vol. 12, pp. 59-68, Jan. 1994.