

New Models and Algorithms in Telecommunication Networks

Hamed Pouya

A Thesis

In the Department

of

Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of

Doctor of Philosophy (Computer Science) at

Concordia University

Montréal, Québec, Canada

November 2018

© Hamed Pouya, 2018

CONCORDIA UNIVERSITY
SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: Hamed Pouya

Entitled: New Models and Algorithms in Telecommunication Networks

and submitted in partial fulfillment of the requirements for the degree of

Doctor Of Philosophy (Computer Science)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____ Chair
Dr. Luis Amador

_____ External Examiner
Dr. Jacques Desrosiers

_____ External to Program
Dr. Ivan Contreras

_____ Examiner
Dr. Thomas Fevens

_____ Examiner
Dr. Jaroslav Opatrny

_____ Administrative Supervisor
Dr. Chadi Assi

Approved by _____
Dr. Volker Haarslev, Graduate Program Director

January 18, 2019 _____
Dr. Amir Asif, Dean
Gina Cody School of Engineering and Computer Science

Abstract

New Models and Algorithms in Telecommunication Networks

Hamed Pouya, Ph.D.

Concordia University, 2018

The telecommunications industry is growing very fast and frequently faces technological developments. Due to the competition between service providers and high expected reliability from their customers, they should be able first, to migrate their networks to the novel advancements in order to be able to meet their customers' latest requirements and second, to optimally use the resources in order to maximize their profitability.

Many researchers have studied different scenarios for Network Migration Problem (NMP). In these studies, a comparison between the legacy and new technologies is investigated in terms of time frames, reduction in expenditures, revenue increases, etc. There have been no prior studies considering the operational costs of NMP e.g., technicians, engineers and travels. The first contribution of the thesis is to propose a two-phase algorithm based on the solution of column generation models that builds a migration plan with minimum overall migration time or cost.

The second contribution is an improved decomposition model for NMP by removing the symmetry between the network connections. We apply a branch-and-price algorithm in order to obtain an ε -optimal ILP solution.

The third contribution of the thesis is to propose a new methodology for Wavelength Defragmentation Problem to recover the capacity of WDM networks in dynamic environments and optimize resource usages. Since rerouting the lightpaths in an arbitrary order may result in a huge number of disruptions, an algorithm based on a nested column generation technique is proposed. The solution is an optimized configuration in terms of resource usage (number of links) that is reachable by no disruptions from the current provisioning.

All the algorithms presented in this thesis are based on Column Generation method, a decomposition framework to tackle large-scale optimization problems.

Acknowledgments

First of all, I am grateful to professor Chadi Assi without whom I would not be able to finish my Ph.D. I'd also like to thank professor Brigitte Jaumard, for her guidance and financial support during my studies. I am deeply grateful to Mitacs and Ciena (Montreal) that provided the opportunity of investigating real world problems during my 3-year fellowship in Ciena. I have to thank Rami Fahim and Marc-Antoine de Santis as my advisors in Ciena for all the things they taught me and all their support.

I am grateful to my parents, my brothers and my sister for their encouragements throughout the years of my study. I am also grateful to my daughter, Delaram, whose smiles made the last year of this long journey sweet. And finally, last but by no means least, to my beloved Maryam that I have always felt her warm presence and constant support. There is no way I can express my gratitude to her in words.

Contents

List of Figures	ix
List of Tables	xi
Chapter 1 Introduction	1
1.1 The Studied Problems	1
1.1.1 Network Migration Problem	2
1.1.2 Wavelength Defragmentation Problem	2
1.2 Plan of Thesis	3
Chapter 2 Technology Background	5
2.1 Optical Networks	5
2.2 SONET	6
2.2.1 SONET Signal	7
2.2.2 SONET Devices	8
2.2.3 SONET Network	9
2.3 Wavelength Division Multiplexing	9
2.3.1 WDM Transponder	10
2.3.2 Wavelength Selective Switch (WSS)	11
2.3.3 Splitter	11
2.3.4 Multicast Switch (MCS)	12
2.4 Software Defined Optical Networks (SDON)	13
2.4.1 Optical Add/Drop Multiplexer (OADM)	14
2.4.2 Reconfigurable Optical Add/Drop Multiplexer (ROADM)	15

2.4.3	CDC ROADM	16
Chapter 3 Migration Plan with Minimum Overall Migration Time or Cost		19
3.1	Introduction	19
3.2	SONET/SDH Network Migration: Problem Statement	22
3.3	Network Migration Optimization Models	24
3.3.1	Notations and Variables	25
3.3.2	Technician Shift Configuration	26
3.3.3	ILP Formulation of STP Phase	28
3.3.4	ILP Formulation of MTP Phase	31
3.4	Solution Process	33
3.4.1	Column Generation and Integer Solutions	33
3.4.2	STP Configuration Generator	35
3.4.3	MTP Configuration Generator	36
3.4.4	Overall View of the Solution Process	40
3.5	Numerical Results	41
3.5.1	Data Sets	42
3.5.2	Efficiency of the Solution Process	42
3.5.3	Migration Cost Analysis	45
3.5.4	Migrated Circuits in STP vs. MTP	47
3.5.5	Shift Usage	47
3.5.6	Travel cost	48
3.6	Conclusions	49
Chapter 4 Network Migration Problem: A New Application for Synchronized Vehicle Routing Problem		50
4.1	Introduction	50
4.2	Problem Statement	55
4.2.1	Problem Description	55
4.2.2	Complexity	56
4.3	Problem Formulation	57

4.3.1	Parameters	59
4.3.2	Master Problem	60
4.3.3	The Pricing Problem: An ILP Shift Generator	63
4.4	Branch-and-Price Algorithm	65
4.4.1	Branching Strategy	65
4.4.2	Strengthening the Lower Bound	67
4.5	Heuristic for Deriving a Scheduling Solution from a Planning Solution . . .	70
4.6	Numerical Results	71
4.6.1	Data sets	72
4.6.2	Performance and Efficiency of the Solution Method	73
4.6.3	Network Migration Cost Analysis	75
4.7	Conclusions	79
Chapter 5 Wavelength Defragmentation for Seamless Migration		80
5.1	Introduction	80
5.2	Literature Review	82
5.2.1	WDM Network Reconfiguration	82
5.2.2	Routing and Wavelength Assignment	85
5.3	WDM Network Reconfiguration Problem: Our Framework	85
5.3.1	Overview of our WDM Network Reconfiguration Framework	86
5.3.2	Triggering of the Wavelength Defragmentation	88
5.3.3	Routing and Wavelength Assignment	88
5.4	A Nested Decomposition Wavelength Defragmentation Algorithm	89
5.4.1	Dependency Graph and Lightpath Rerouting Order	90
5.4.2	Identification of Avoidable Disruptions: Reducing the Number of Cy- cles in the Dependency Graph	93
5.4.3	Comparison of the RWA Decomposition Models	94
5.4.4	WDF_MBB Model	97
5.4.5	WDF_NCG Algorithm	98

5.5	Solution of WDF_MBB Model: A Nested Decomposition Wavelength Defragmentation Model	99
5.5.1	Nested Column Generation	99
5.5.2	Re-Establishment of a Feasible RWA Provisioning	102
5.6	Computational Results	103
5.6.1	Traffic and Network Data Sets	103
5.6.2	Generation of Fragmented RWA Provisionings	104
5.6.3	Reduction of the Number of Rerouting Deadlocks	107
5.6.4	Seamless or Almost Seamless Wavelength Defragmentation	108
5.6.5	Number of Reroutings vs. Bandwidth Improvement	111
5.7	Conclusions	113
	Chapter 6 Conclusions and Future Work	114
	Bibliography	116
	Appendices	128
	Appendix .A Generic Pricing Problem (GPP) for Model in Section 4.4.2	128
	Appendix .B Pricing Problem (PrPP) Using a Set of Precomputed Shifts for Model in Section 4.4.2	130

List of Figures

Figure 2.1	Part of a simple network using SONET equipment	9
Figure 2.2	A transponder	11
Figure 2.3	A $1 \times N$ WSS	11
Figure 2.4	A passive splitter	12
Figure 2.5	A 2×3 multicast switch	12
Figure 2.6	SDON architecture	14
Figure 2.7	OADM at a degree-two node	15
Figure 2.8	ROADM at a degree-two node	16
Figure 2.9	Add direction of a contentionless ROADM	17
Figure 3.1	Example of a legacy network and its associated set of technicians . .	23
Figure 3.2	Example of four technician shift configurations	27
Figure 3.3	Circuits migrated in STP vs. MTP	29
Figure 3.4	CG ILP algorithm	34
Figure 3.5	Proposed two-phase algorithm	41
Figure 3.6	Circuit distribution between pairs of sites	45
Figure 3.7	Shift usage vs. cost per circuit	48
Figure 4.1	An example of a telecommunication network for network migration problem	51
Figure 4.2	An example of reducing an instance of partition problem to an NMP instance	58
Figure 4.3	A possible solution as a subset of shifts	58

Figure 4.4	Algorithm for strengthening the lower bound	69
Figure 4.5	Migration cost vs. shift usage	76
Figure 4.6	Cost analysis of \bar{n}^{CIR} on real data set	76
Figure 5.1	WDM Network Reconfiguration Process	87
Figure 5.2	An Example of Dependency Graph	91
Figure 5.3	Preprocessing Operation	94
Figure 5.4	Nested Column Generation	101
Figure 5.5	Fragmented Networks (GoS)	105
Figure 5.6	Performance of Algorithm 3: Reduction of the size of the biggest strongly connected component.	108
Figure 5.7	Largest SCC for USA 5%	109
Figure 5.8	Percentage of Shortest Paths	112
Figure 5.9	Rerouting vs. Bandwidth Improvement	112

List of Tables

Table 2.1	SONET rates	7
Table 3.1	Data sets	42
Table 3.2	Migration costs	43
Table 3.3	Efficiency of the algorithm	44
Table 3.4	Average unit circuit cost (\$)	46
Table 3.5	Number of circuits migrated in STP and MTP	47
Table 3.6	Travel cost (\$)	49
Table 4.1	Artificial data set	72
Table 4.2	Real data set	73
Table 4.3	Algorithm performance on artificial data set	74
Table 4.4	Algorithm performance on real data set under 300 seconds time limit	75
Table 4.5	Cost analysis on real data set	78
Table 5.1	List of Dependencies	92
Table 5.2	Comparison of 4 decompositions	96
Table 5.3	Main characteristics of the networks	104
Table 5.4	Statistics for the Characteristics of the Strongly Connected Components in the Dependency Graphs	104
Table 5.5	Statistics for the SCCs after preprocessing algorithm	106
Table 5.6	Performance of the WDF_NCG algorithm	110
Table 5.7	Bandwidth requirement compromise for a make-before-break reachable optimized wavelength provisioning	111

Table 5.8 Length Reduction for USA	111
--	-----

Chapter 1

Introduction

Section 1.1 briefly introduces the problems studied in this thesis. The plan of thesis is presented in Section 1.2.

1.1 The Studied Problems

In this thesis, we study two critical problems in telecommunications industry. Telecommunication is a fast growing industry entailing regular technology developments. Communication Service Providers (CSPs) have to adapt to the latest technologies by upgrading their networks. For instance, most CSPs are expected to operate SONET/SDH infrastructures until the end of this decade (2020), before ultimately migrating to IP-centric, Software Defined Optical Networks (SDON) while SONET/SDH was originally architected in the early 1990s [64]. Although such a migration is a major undertaking, it provides great saving opportunities e.g., 30% reduction in network congestions and 50% reduction in CAPEX [64]. CSPs need effective optimization algorithms in order to predict the time and resources required for the migration of large networks, which can stretch up to several years.

After migrating to the new technology, network operators face the second problem that is how to maximally use the resources. The dynamic environment in telecommunications and the rapid increase in traffic demand show the importance of efficient resource usage in optical networks. Since the current state of the network is considered in order to fulfill any

new demands, networks frequently become fragmented and need to be reoptimized. Network operators require optimal solutions in terms of resources that result in minimum disruptions and can be obtained in short time. Defragmentation can result in 75% bandwidth recovery in some cases [9].

1.1.1 Network Migration Problem

Network Migration Problem (NMP) arises when network operators need to upgrade the network infrastructure. One of the challenges faced by the operators of telecommunication networks is to upgrade their infrastructure to support increasing capacity demands driven by the high-bandwidth application needs of end-user consumers and businesses. While this modernization is inevitable, it can take from several weeks to even a few years, require different resources and have operational impacts (outages) on the network. Hence, it is critical for the network operators to ensure that network migration is completed on time, with minimal outages and minimum costs.

In a network migration problem, the goal is to design a migration plan with minimum overall migration time or cost, i.e., to develop an optimization framework for planning the workforce (number of technicians/engineers) and the time (duration of the migration). This planning minimizes the cost of migration defined in terms of the costs associated with required workforce and time.

1.1.2 Wavelength Defragmentation Problem

In telecommunication networks, data is transferred by means of wavelength over a path from the source to the destination. The combination of a wavelength and a path is called a *lightpath*. In dynamic environments, lightpaths are needed for a short period of time and demands are not known in advance. Hence, the demands are fulfilled based on the current state of the network with no knowledge about the upcoming requests. In this situation, after dynamically fulfilling the demands, the network becomes fragmented i.e., there is enough capacity in the network but it is stranded and there is higher chance that new requests are denied. In order to make room for new requests and recover the stranded

bandwidth, we may need to change the lightpaths assigned to current demands. Due to technical and capacity constraints, an arbitrary order of changing the lightpaths may result in a great number of disruptions. *Wavelength Defragmentation Problem* studies the process of migrating from the current fragmented state of the network to an optimized state in most seamless possible fashion.

1.2 Plan of Thesis

In Chapter 2, a review on the basic concepts of optical networks is presented. These concepts are used widely in the thesis. In Chapter 3, we introduce network migration problem. We propose an optimization framework that builds a migration plan with minimum overall migration time or cost, i.e., estimates the number of required maintenance windows, engineers and technicians in order to perform the network migration. Network migration problem is formulated as a set of circuit migration problems, each requiring technician synchronization in order to minimize the outages. We propose a two-phase algorithm, that relies on the solution of column generation models, in order to provide accurate migration plans for realistic size networks. Extensive numerical experiments are conducted on several real network migration data instances arising from customers of Ciena.

In Chapter 4, we propose a more scalable formulation (than the model proposed in Chapter 3) for NMP by removing some symmetries between the circuits. We use column generation technique and design a branch-and-price algorithm in order to guarantee the quality of ILP solution. The algorithm is tested on several real and artificial data sets.

In Chapter 5, we study a wavelength defragmentation problem in SDONs. Due to the dependencies between the current configuration of the network and the optimum configuration, the lightpaths in the new configuration and order of rerouting them play an important role in breaking the dependencies. We propose a methodology to recover and reoptimize the stranded bandwidth of optical networks with no disruptions by rerouting the lightpaths corresponding to the demands. The goal is to find a configuration as close as possible to the optimum solution and reachable by no disruptions. The solution scheme (*i*) computes an

optimal lightpath provisioning, *(ii)* checks if it is reachable by no disruption from the current fragmented one, *(iii)* if not, goes on with an iterative process which recomputes a lightpath provisioning subject to additional constraints for eliminating the rerouting deadlocks in order to define migration without disruptions. The resulting wavelength defragmentation process is thoroughly tested on various data and network instances.

Chapter 2

Technology Background

In this chapter, we will introduce some of the basic concepts that are the foundations of optical networks. These concepts are widely used in this thesis. We will review the evolution of optical networks in Section 2.1. Since at the current stage of technology, network migration problem (Chapters 3 and 4) is mostly applicable to SONET networks, fundamentals of this standard is presented in Section 2.2. Sections 2.3 and 2.4 introduce Wavelength Division Multiplexing and Software Defined Optical Networks, respectively. Wavelength defragmentation problem (Chapter 5) is a natural byproduct of these two technologies.

2.1 Optical Networks

An optical network consists of fiber links combined with equipment which are capable of processing light. The advent of optical networks is the result of network migrations. In 1980s, carriers began migrating physical layer of their networks to fiber optic cables. Optical fiber is a lightweight cable which provides more clear transmissions with lower loss and higher potential capacity in comparison to older copper cables [87]. While discussing about optical networks, we talk about two generations of optical networks. In the first generation, light was utilized to provide more capacity and all intelligent network operations were performed by electronics. Second generation of optical networks handles routing, switching and intelligence in the optical domain [78].

Simmons [87] defines the value of an optical network as a function of capacity, reliability, cost, scalability, and operational simplicity. That is why companies always look for new technologies and as a result, technological developments occur very fast. *Wavelength Division Multiplexing (WDM)* is one of the first technology improvements in optical networks. This technology refers to the ability of carrying multiple light channels on one fiber. These channels (wavelengths) which are carried on different frequencies are combined together into a single fiber. The other technological advancement that made the networks more cost effective was the development of the *Erbium-Doped Fiber Amplifier (EDFA)*. Before EDFAs, each wavelength had to be regenerated at about every 40 km to restore the signal quality. EDFA deployment allowed signals to be transmitted more than 500 km without regeneration. Recent EDFA improvements have increased this distance to 1,500 – 2,500 km [87, 78]. Due to technological changes, optical networks are very different from initial days in terms of efficiency and costs. In 1990s, the maximum bit rate was no more than 2.5 Gb/s (1 gigabit is 10^9 bits) while it is now 10 to 100 Gb/s and the deployment of 400 Gb/s to 1 Tb/s is expected in the 2015 – 2020 time frame [87].

Fast technological improvements in addition to increasing traffic demand and new customer requirements make companies optimize and reoptimize the resources and finally adapt their networks in order to remain competitive in the industry. Companies encounter a re-configuration problem while trying to reoptimize their networks and a migration problem while adapting themselves to a new technology.

2.2 SONET

Today's high data rate technologies like video conferencing require high bandwidth of optical fibers. The network with wide bandwidths for transmission and equipment sold by different manufacturers requires a single clock to cope with timing of transmissions and equipment deployed in the network. *Synchronous Optical Network (SONET)* is a standard with a powerful frame design to carry signals which provides recommendations for the standardization of *Fiber-Optic Transmission Systems (FOTS)* equipment [36]. SONET

facilitates an optical telecommunication transport and provides a large set of operational parameters for optical transmission systems. This standard was developed by *Exchange Carriers Standards Association* (ECSA) for *American National Standards Institute* (ANSI) [7]. SONET is an example of *Time-Division Multiplexing* (TDM) system in which the bandwidth of the fiber is considered as a channel. This channel is divided into time slots to create subchannels. The transmission of bits in this synchronous system is controlled by a master clock with high level of accuracy [36].

2.2.1 SONET Signal

SONET uses an optical hierarchy to carry a lot of signals from different sources and at different capacities which supports a certain data rate specified in megabits per second (Mbps). The base of a SONET signal is called *Synchronous Transport Signal Level 1* (STS-1) [7]. The corresponding optical signals are called *Optical Carriers* (OCs) [36]. These data rates are represented in Table 2.1.

Table 2.1: SONET rates

STS	OC	Rate (Mbps)
STS-1	OC-1	51.840
STS-3	OC-3	155.520
STS-9	OC-9	466.560
STS-12	OC-12	622.080
STS-18	OC-18	933.120
STS-24	OC-24	1244.160
STS-36	OC-36	1866.240
STS-48	OC-48	2488.320
STS-192	OC-192	9953.280

As shown in Table 2.1, the STS-9 is exactly nine times STS-1, three times STS-3 and one-half STS-18. This means that 18 STS-1 or 6 STS-3 can be multiplexed into one STS-18, and so on.

Traffic generated by a customer (voice, data, video, etc.) across the circuits in the network is carried at a digital rate which is called *Digital Signal level 1* (DS-1). This traffic is therefore called a DS-1 which operates at rate of 1.544 Mbps. Higher-speed communications can be achieved by multiplexing four DS-1s into one DS-2. It is still possible to go further

and achieve higher-speed communications by multiplexing seven DS-2s. DS-3 supports the 44.736 Mbps rate of speed and is used by public carriers (long distance carriers) [7]. STS-1 is actually designed to accommodate data rates equivalent to those of the DS-3. The difference in the capacity of DS-3 and STS-1 is provided to handle the overhead needs of the optical systems [36].

2.2.2 SONET Devices

SONET transmission relies on three basic devices which are STS multiplexer/demultiplexer, regenerators, add/drop multiplexers [36].

STS Multiplexer/Demultiplexer

This device is considered as the beginning and end point of a SONET link. An STS multiplexer multiplexes signals from electrical domain and creates the corresponding OC signal. An STS demultiplexer acts in the opposite direction and converts an optical OC signal into corresponding electrical signals.

Regenerator

Regenerator extends the length of a link. This device operates in the physical layer. Signals that carry information within a network can travel a fixed distance before attenuation endangers the integrity of the data. A regenerator receives an optical signal (OC-n) and, before it becomes too weak or corrupted, demodulates it into the corresponding electrical signal (STS-n) and regenerates the electrical signal. Finally, the electrical signal is modulated into its corresponding OC-n signal. The regenerator then sends the refreshed signal.

Add/Drop Multiplexer

Signals can be inserted or extracted by an add/drop multiplexer (ADM). An ADM can add STSs from different sources into a given path. It also can remove and redirect a signal without demultiplexing the entire signal.

2.2.3 SONET Network

Figure 2.1 represents a part of a SONET network. Each terminal in Figure 2.1, refers to a device which uses the service of a SONET network. As mentioned before, STS multiplexer/demultiplexer is located at the beginning and end of a SONET link. The term on each link (OC-n) shows the rate of that link to transmit data. Each colorful dotted line represents a circuit in Figure 2.1. A circuit begins from one point and may pass through different network devices before reaching the destination.

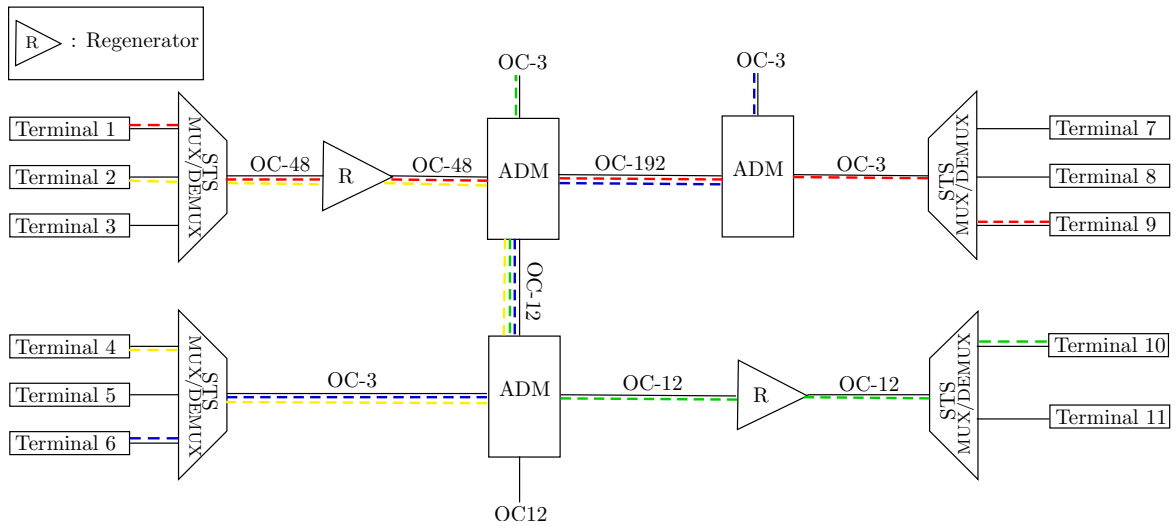


Figure 2.1: Part of a simple network using SONET equipment

In the simple configuration shown in Figure 2.1, incoming signals are fed into a STS mux/demux which combines them and creates an optical signal. This signal might be transmitted to a regenerator where it is recreated without noise. When a signal is fed into an ADM, a desired signal might be inserted or extracted from it. ADM, then, recognizes these signals and, if necessary, redirects and sends them to a given path for each signal.

2.3 Wavelength Division Multiplexing

There are two ways to increase the capacity of data transmission on a fiber. Increasing the bit rate is the first option. This means that many lower-speed data streams are multiplexed into a higher-speed stream by means of TDM. For example, by picking 1 byte of data

from each stream in each time slot, thirty two 155 Mb/s streams can be multiplexed into a 5 Gb/s stream. The highest reachable transmission rate by TDM is 40 Gb/s. The second way to increase the capacity is WDM. The idea behind WDM is to transmit data simultaneously over multiple wavelength (frequencies). The wavelengths are selected sufficiently far from each other to avoid interference i.e., two signals cannot be on the same wavelength. In other words, WDM provides some virtual fibers over a single link and each fiber carries a single data stream. Early WDM systems could support at most 10 wavelengths but this number is now over 100 wavelengths per fiber. WDM systems are now widely used in long-haul and metro networks. It should be mentioned that TDM and WDM are complementary to each other and the networks today deploy a combination of both.

Every wavelength needs to be cleaned up or *regenerated*. The maximum distance a wavelength can travel before it needs to be regenerated is called *optical reach*. Every connection transmitted on a wavelength might go through several nodes on its path. If the connection needs to stay on the same wavelength from the source to the destination, we face a *wavelength continuity constraint*. Considering this constraint and the fact that the same wavelength cannot serve more than one signal on a link, wavelength assignment to connections on every link affects the assignments of the wavelengths on other links. Note that it is possible to deploy converters to change the wavelength of a connection i.e., the connection enters and exits a node on different wavelengths. However, due to high cost of converters, they are impractical and wavelength continuity constraint remains a relevant factor [78, 87].

In next sections, we introduce some of the most important components of WDM networks. In Section 2.4, we will see how these equipment help us design more flexible networks.

2.3.1 WDM Transponder

A WDM transponder takes a signal from a client of a network and converts it to the electrical domain via an interface. The electrical signal drives a WDM-compatible laser to convert the electrical signal to a particular wavelength. Transponders might be fixed or tunable. In fixed transponders, client signal can be converted to just one particular

wavelength while in tunable transponders, the client signal can be converted to a range of optical frequencies. In the reverse direction, WDM-compatible signal is converted to the client signal. Figure 2.2 illustrates a transponder.

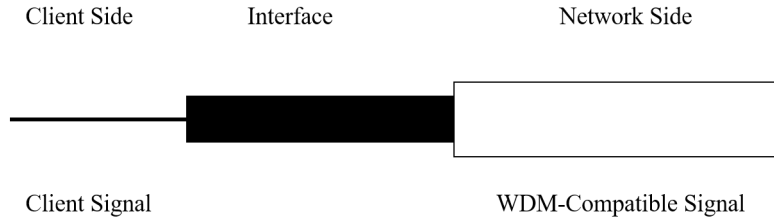


Figure 2.2: A transponder

2.3.2 Wavelength Selective Switch (WSS)

A WSS is capable of treating each wavelength of a WDM signal differently. A $1 \times N$ WSS has a single port on one side and N ports on the other side. A WSS can direct any wavelength from input port(s) to any of the output port(s). Figure 2.3 illustrates a $1 \times N$ WSS.

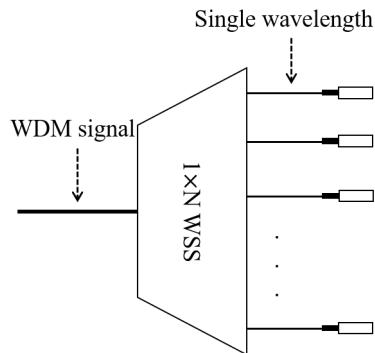


Figure 2.3: A $1 \times N$ WSS

2.3.3 Splitter

A splitter is a passive device that acts as a splitter in receive direction and as a coupler in the transmit direction. The received signal is split (not multiplexed) and transmitted to the transponders. This means that if the input signal is WDM, the output signals are also

WDM. Every splitter has one input port and N output ports. Splitters are mostly deployed in the architectures designed to broadcast the signals. Figure 2.4 illustrates a passive splitter.

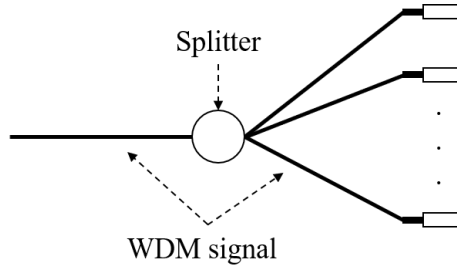


Figure 2.4: A passive splitter

2.3.4 Multicast Switch (MCS)

An MCS is an alternative for WSS as it is cheaper and has smaller physical size. However, an MCS is not a wavelength selective device e.g., if the input signal is WDM, the output signal is also WDM. An MCS directs an incoming WDM signal to the transponders corresponding to the connections contained in that signal. Figure 2.5 illustrates a 2×3 multicast switch. As seen in Figure 2.5, every incoming signal is multicast to corresponding transponders e.g., incoming east WDM signal is multicast to east and north transponders. Transponders in an MCS are equipped with frequency-selective filters to select the desired wavelength from the WDM signal.

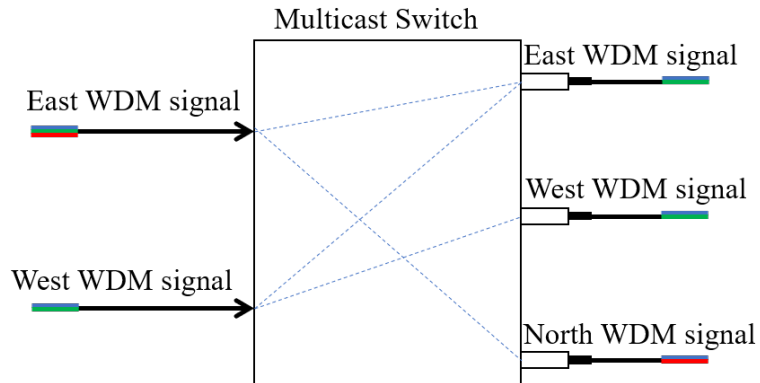


Figure 2.5: A 2×3 multicast switch

2.4 Software Defined Optical Networks (SDON)

The rise of cloud services, using mobile devices and on-demand services e.g., a service for a period of time, have changed the traffic patterns. As a result, networks have to become more flexible to fulfill these needs meaning that engineers can alter network behavior in real-time and deploy new applications and network services in a matter of hours or days, rather than the weeks or months needed today. Today's networks are highly vendor dependent. Implementing a network-wide policy requires to configure thousands of devices, while deploying new capabilities in each component of the network is hindered by vendor's product cycles. For example, routers and switches are usually "close" systems, often with limited and vendor-specific control interfaces. Therefore, once deployed and in production, it is quite difficult for current network infrastructure to evolve i.e., deploying new protocols. It should be considered that the networks grow as the demands rapidly grow. This is equal to the addition of hundreds and thousands of devices that must be configured and managed.

The idea behind programmable networks is to overcome these difficulties and make network operations faster and more flexible. Software Defined Networks (SDN) or Software Defined Optical Networks (SDON) is an emerging platform where network control is directly programmable. In SDON, the network intelligence is centralized in software-based controllers (the control layer in Figure 2.6). Centralizing the intelligence in the control layer gives the network the capability of (re)configuring, managing, securing, and (re)optimizing network resources via dynamic, automated SDON programs. With SDON, enterprises and carriers obtain vendor-independent control over the entire network that simplifies both network operations and network devices themselves as they no longer need to understand and process thousands of protocol standards but merely accept instructions from the SDON controllers (the infrastructure layer in Figure 2.6).

SDON architectures support a set of *Application Programming Interfaces* (API) that make it possible to implement common network services, including routing, multicast, security, access control, bandwidth management, traffic engineering, quality of service, processor and storage optimization, energy usage, and all forms of policy management, custom

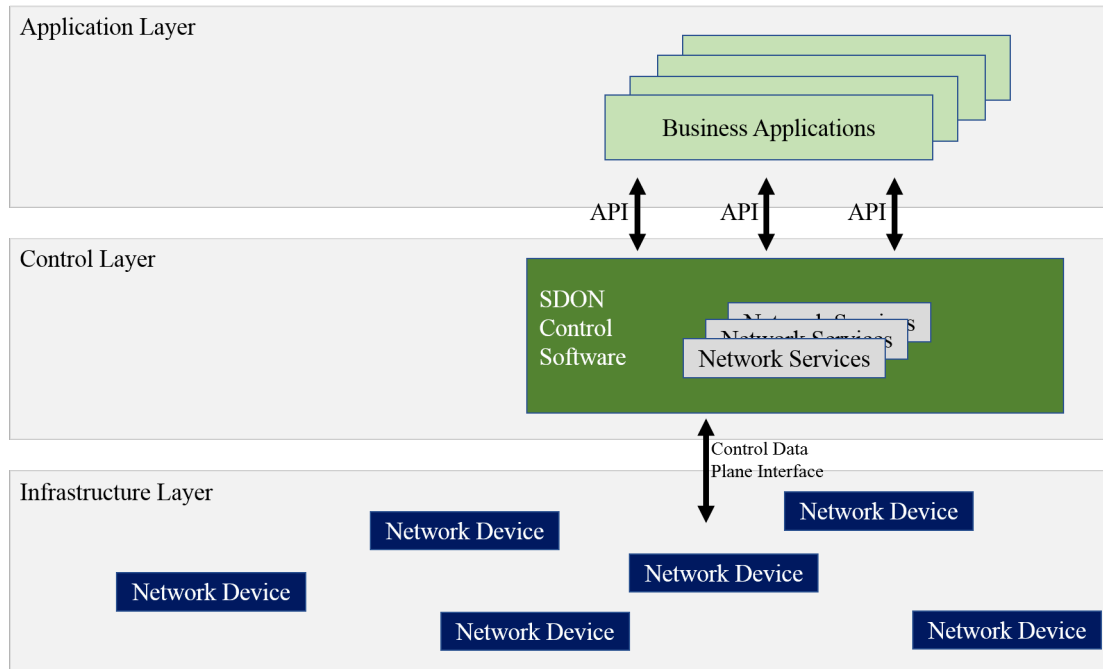


Figure 2.6: SDN architecture

tailored to meet business objectives [81, 71, 3, 38].

Since most of the operations in SDNs require frequent adding/dropping and (re)routing the demands in real time, in next sections, we will investigate the most common component used in SDNs for these functionalities.

2.4.1 Optical Add/Drop Multiplexer (OADM)

Although *Optical Add/Drop Multiplexer* (OADM) have been available from the mid-1990s, significant deployment did not start before 2000. The name of the equipment comes from SONET/SDH add/drop multiplexer (ADM), which is capable of adding/dropping lower-rate SONET/SDH signals to/ from a higher-rate signal without terminating the entire higher-rate signal. Similarly, the OADM adds/drops wavelengths to/from a fiber without having to electronically terminate all of the wavelengths comprising the WDM signal. Figure 2.7 illustrates a degree-two OADM. As seen in Figure 2.7, WDM signals from two directions east and west enter and exit the node. In inbound directions, the WDM signals

are demultiplexed and in outbound directions, they are multiplexed. Each individual wavelength is directed either to another direction if the traffic is *passthrough* or is dropped at one of the drop ports. The add signals are also directed to one of the outbound directions.

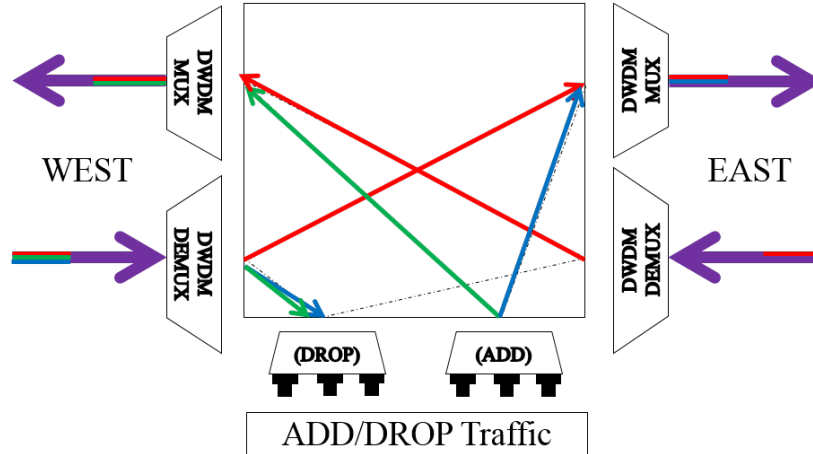


Figure 2.7: OADM at a degree-two node

The most important property of OADMs is their degree of reconfigurability. Earliest OADMs had a fixed switching matrix, meaning that it needed to be specified which particular wavelength would be added/dropped at each node while the others would transit the node as passthrough traffic. After installation, these OADMs were fixed in their configurations. This property highly limits the capability of the network to adapt dynamic traffic patterns. It is clear that such an architecture is not consistent with the dynamic of the services provided via SDON.

2.4.2 Reconfigurable Optical Add/Drop Multiplexer (ROADM)

A reconfigurable OADM is called ROADM. Nowadays, most of the OADMs are configurable. This implies that any wavelength can be added or dropped at any nodes and changing the choice of the wavelength to be added or dropped does not affect the other terminating or transiting connections at that node. The biggest limit of the ROADMs is the threshold of add/drop traffic. A typical threshold is having the capability to add/drop 50% of the supportable wavelengths. The ROADMs that support more than 50% add/drop are expensive due to the amount of required transponders.

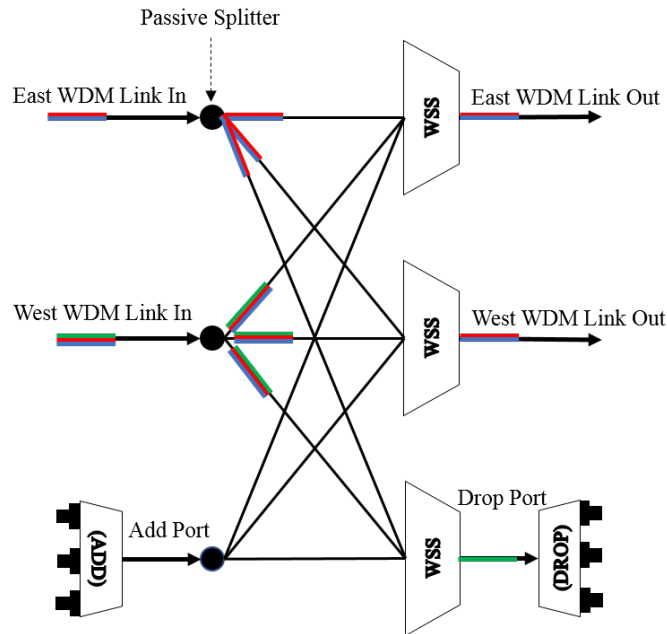


Figure 2.8: ROADM at a degree-two node

Figure 2.8 illustrates a degree-two ROADM. Every inbound link and every add port is connected to a passive splitter that transmits a copy of the WDM signal to each of the outbound directions and the drop port. The WSS devices connected to the outbound links and the drop port select the desired wavelengths and block the rest. As seen in Figure 2.8, every wavelength can be added or dropped at any node using this architecture.

2.4.3 CDC ROADM

Colorless, Directionless and *Contentionless* are the most important properties of a ROADM. A ROADM with these three properties is called CDC ROADM. Being colorless shows that every slot of a ROADM can accommodate a transponder of any wavelength. Due to wavelength continuity constraints, it is important to use a particular wavelength for a specific connection. If the ROADM is colorless, it gives us the flexibility to insert a tunable transponder into any of its slots and then tune the transponder to the corresponding wavelength. Directionless refers to the ability of the ROADM to route a connection from a transponder to any of the links. The architecture in Figure 2.9 is directionless as all transponders are connected to all links. A contentionless ROADM does not block a

connection just because of the contention within the ROADM. In other words, a ROADM is contentionless if it can establish a connection over an available wavelength on a link. The architecture in Figure 2.8 is not contentionless. Assume that we need to add two connections, one on east link and one on west link. Since the only wavelength available on both links is the green wavelength, both connections should be established on this wavelength. This means that the WDM signal exiting the add port has to carry two signals both on the same wavelength which is not possible and at least one of the connections has to be blocked so that the other one can be routed. The same thing might also happen in a drop port and two connections on the same wavelength need to be dropped. In order to make the architecture shown in Figure 2.8 contentionless, we have to add more ports and more transponders that will increase the cost.

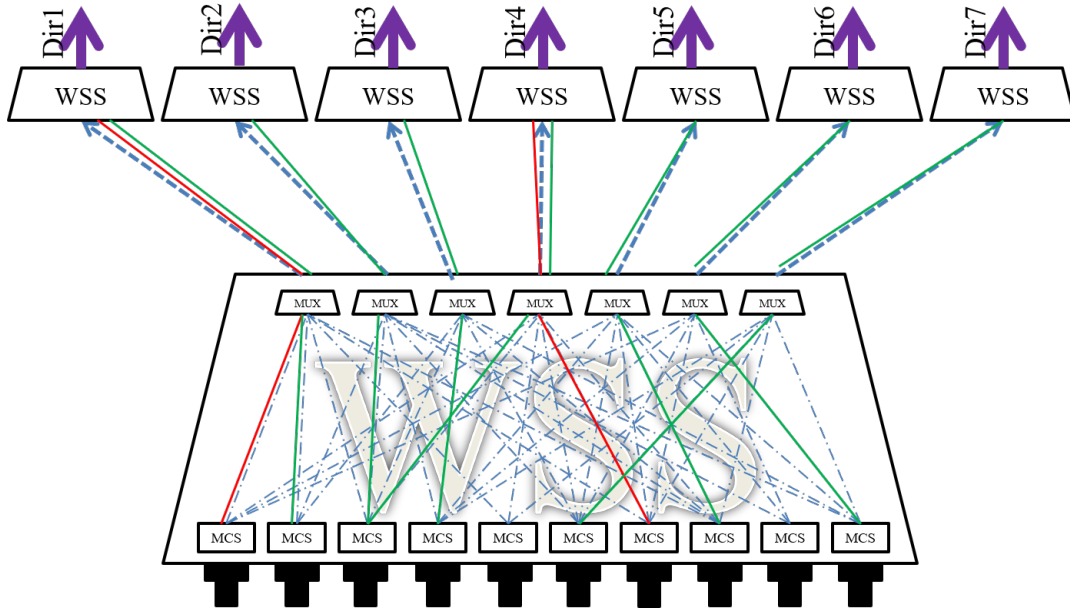


Figure 2.9: Add direction of a contentionless ROADM

Figure 2.9 shows the architecture of an add direction in a contentionless ROADM. As seen in Figure 2.9, every transponder is connected to an MCS that multicasts a wavelength to corresponding outgoing links. All wavelength received at each link are multiplexed and then selected by WSS according to their routings. Figure 2.9 shows that every transponder has access to all links via all wavelengths e.g., green wavelength. In this case, if a wavelength

is available on a link, the ROADM does not block any connections due to internal contention.

Considering the capability of ROADMs to be remotely configurable as opposed to require manual interventions, they are among the most important components of SDONs. They allow the engineers to turn up or terminate the lightpaths frequently, reconfigure the networks and reoptimize the resources with high flexibility that is actually the stem of wavelength defragmentation problem and the solutions provided.

Chapter 3

Migration Plan with Minimum Overall Migration Time or Cost

3.1 Introduction

Network operators are today challenged with upgrading their SONET/SDH infrastructure to support increasing capacity demands driven by the high-bandwidth application needs of end-user consumers and businesses. However, such a network modernization is often characterized by a lack of critical in-house resources in order to ensure that migration projects are completed on time, with minimal operational impact (outages) and at minimum cost [18], [22]. In addition, network migration, which results from a network modernization, can be spanned over a long period of several months, even a few years, see, e.g., the Telstra network migration to an IP/MPLS network core which took 18 months, with a significant network simplification of 220 enhanced nodes replacing the 856 nodes that existed previously [23].

Network migration arises in very different contexts. While in this paper, we are interested in network migration in the context of network modernization, network migration may result from less drastic network changes, and be more gradual, e.g., in the context of capacity exhaustion. In this last case, network migration follows some capacity upgrade

such as increase of the number of channels/fibers, increase of the bite rate, of some wavelengths, see [61]. Network migration also occurs in the context of network reconfigurations, which are conducted in responses to changes of upper-layer traffic, network failures, or new deployment of network resources [63]. More recently, several work deal with virtual network migration, which is the process of remapping a logical topology to a new set of physical resources to provide, e.g., improved resource utilization, failure recovery, energy savings, or defense against attack [67, 34].

In the context of network modernization, while various recent white papers [37, 22, 1] address network modernization in order to enhance service performance, no studies are available on how to accurately estimate the cost and the time of a network migration. Türk *et al.* [96] focus on discussing a cost model for network migration, which includes Capital Expenditures (CAPEX), Implementation Expenditures (IMPEX) and Operational Expenditures (OPEX).

Jaeger and Huelsermann [42] perform a cost analysis of network migration from SDH network to a packet based transport network, comparing two scenarios, one in which all SDH services remain in the SDH platform until their normal end-of-life and another scenario in which all SDH services are emulated in the packet platform. Podhrasky [74] presented a short description of different migration scenarios toward Next Generation Networks (NGNs).

Wavelength-Division Multiplexing (WDM) networks have an extra degree of difficulty in the network migration process. Since the fibers in these networks may be related to different optical connections, outage of a single link may lead to problems in multiple connections. Bley *et al.* [10] studied migration problem in a WDM network in order to minimize the total number of disruptions.

The objective of this study is to design a migration plan with minimum overall migration time or cost, i.e., to develop an optimization framework for planning the workforce (number of technicians/engineers) and the time (number of required maintenance windows). This planning minimizes the cost of migration defined in terms of the costs associated with required workforce and time.

Such a resource planning problem has strong similarities with the Vehicle Routing Problem (VRP) subject to synchronization constraints. While in classical vehicle routing problems, synchronization is necessary between the vehicles with respect to which vehicle visits which customer, vehicle routing problems with synchronization constraints (VRPS) exhibit additional synchronization requirements with regard to spatial, temporal, and load aspects, e.g., a vehicle routing problem in which more than one vehicle may or must be used to fulfill a task, see, e.g., the recent survey of Drexl [29]. In the context of network migration, a vehicle corresponds to a technician synchronized with another technician such that both technicians simultaneously migrate the two endpoints of a circuit in order to minimize the outage. While VRPS has been studied with only one depot (see, e.g., [11, 84]), network migration involves several sets/pools of technicians (i.e., depots) distributed over several regions, so that technician workforce is more difficult to plan than in the classical workforce problems [13]. The VRPS problem closest to the circuit migration problem is that of snow plowing operations for which Salazar-Aguilar *et al.* [84] proposed a quadratic optimization model and a heuristic, while we are able to formulate the circuit migration problem with a linear optimization model, that can be solved efficiently using column generation techniques. The technician scheduling in network migration problem was first introduced in [46]. The proposed approach is scalable just for small instances.

The current paper extends the results of [46] and is organized as follows. The network migration problem is stated in detail in Section 4.2.1. We propose an optimization model in Section 3.3: it has two variants, which are described in Sections 3.3.3 (simple technician pairing problem) and 3.3.4 (multiple technician pairing problem), respectively. Algorithms for solving the proposed optimization models are discussed in Section 3.4. Numerical results are presented in Section 5.6 on four case studies provided by Ciena. Conclusions are drawn in the last section.

3.2 SONET/SDH Network Migration: Problem Statement

We study the migration problem of a legacy SONET/SDH network as the migration of a very large number of circuits over a new network, e.g., an OTN network. A circuit c corresponds to a path created from a source to a destination in order to transmit traffic. Every circuit can be viewed as a transport equipment that consists of a set of topological links and provisioned cross connects to switch traffic from one node to another.

We assume that the legacy and new networks are located over a set of geographical regions, with each region being defined as a set of sites. Sites belonging to the same region are such that moving from one site to another can be done within a maintenance window. Examples are different floors of the same building or buildings in different areas of a city.

For each region, we are given its set of neighbor regions i.e., the ones reachable within a car drive of few hours (less than a given threshold). Regions within a neighborhood might share their technicians. Travel times for each pair of sites/regions are given. Network migration is usually scheduled on off peak times and/or during a scheduled maintenance window, thus one can assume that travel times between sites and regions also correspond to low road traffic. Maintenance windows are usually during the night, and their length varies from one to the other (typically from one up to eight hours). Since a migration can last from a few weeks to 1 or 2 years, a time interval defined by a set of maintenance windows is defined at the beginning to perform the migration.

In Figure 3.1, there are 5 regions. The Montreal region has 4 sites and a pool of 3 technicians, and if necessary, Montreal technicians can go to Quebec City or vice versa, Quebec technicians can go to Montreal. An engineer synchronizes the circuit migrations arising in Toronto, Oshawa and Montreal.

In every circuit migration from legacy to new infrastructure, it is inevitable to have some outage. In order to minimize the outage duration, each circuit migration requires the coordination of two technicians, one at each circuit endpoint at the time of the migration. The coordination is handled by an engineer. Every engineer can simultaneously take care of more than one circuit migration. As stated above, migration of circuits is operated during

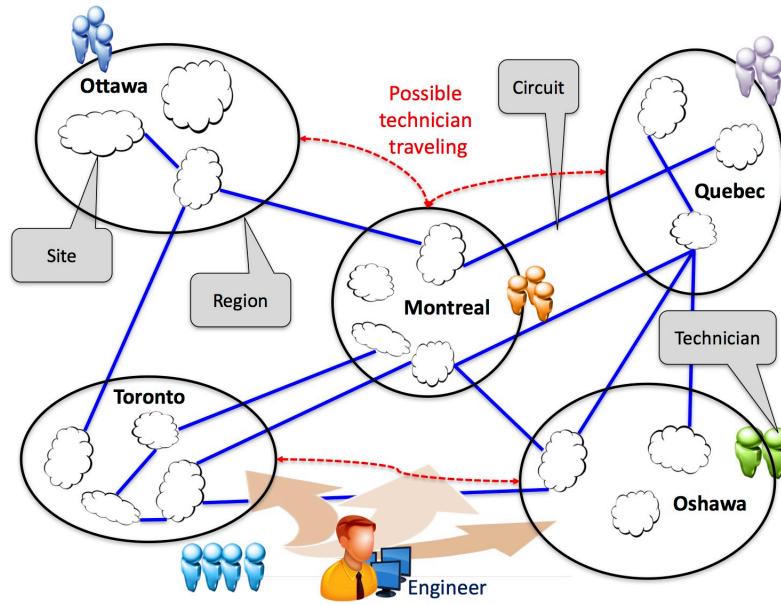


Figure 3.1: Example of a legacy network and its associated set of technicians

low traffic periods which further minimizes the outage for the users. We therefore assume that the customer provides such low traffic maintenance windows, with variable durations.

We assume that there is a pool of technicians in some regions. Regions without a pool of technicians are assumed to be within a neighborhood of a region with at least one technician. Technician shifts may have different durations. Typical values are 6 or 8 hours. Their length should be long enough to cover the time required by the setup and the migration operations during one maintenance window. During every maintenance window, there might be several shifts, each associated with a technician working during the maintenance window.

Given the above input data and constraints, the network migration is defined as:

- Assignment of a set of circuits to each maintenance window for migration,
- Assignment of technicians to the circuit endpoints.

The objective is to minimize the migration cost which is defined as the sum of the technician/engineer salaries, the technician travel costs and the cost of using maintenance windows. Duration is taken into account through the number of required maintenance windows.

3.3 Network Migration Optimization Models

Since the number of circuits may be extremely large, with big bundles of circuits between some pairs of sites, we use a two-phase model as follows.

Simple Technician Pairing (STP) Phase

In the first phase, we detect every pair of sites sharing a large number of circuits such that at least one full shift (6 or 8 hours) is required for their migration. Next, we assign two technicians to each such shift, and assign one technician to each site. In this phase, technicians cannot travel between sites during a shift because they are busy migrating circuits during the entire shift. However, some travel costs might occur if a site is assigned to a technician from another region. The objective here is to choose technicians that require minimum travel cost over minimum number of maintenance windows. It should be noted that in STP, every selected technician is synchronized with exactly one other technician during one maintenance window. That is why this phase is called *simple* technician pairing.

Multiple Technician Pairing (MTP) phase

After migrating circuits in STP, migration time required for the rest of the circuits between every two sites is less than the duration of a full shift. In this case, each technician can travel between different sites and/or be synchronized with more than one technician during the shift. In the second phase, a technician can be paired with more than one technician during a maintenance window, hence the name *multiple* technician pairing, see example of Figure 3.2 for an illustration.

STP and MTP phases are formulated as Integer Linear Programming (ILP) models described in Sections 3.3.3 and 3.3.4, respectively. Common notations and variables of the two models are defined in Section 3.3.1.

3.3.1 Notations and Variables

Mathematical formulations of STP and MTP use the following sets of parameters and variables.

Parameters

V	Set of circuit endpoints (indexed by v)
C	Set of all circuits (indexed by $c = \{v, v'\}$)
W	Set of maintenance windows (indexed by w)
S	Set of sites (indexed by s)
R	Set of regions (indexed by r)
N_r	Set of neighbor regions of region r
S_r	Set of sites located in $\{r\} \cup N_r$
C_r	Set of circuits with at least one endpoint in site s such that $s \in S_r$
V_r	Set of circuit endpoints located in region r
V_s	Set of circuit endpoints located in site s
$C_{ss'}$	Set of circuits between sites s and s'
α_{ENG}	Number of technicians coordinated by one engineer
α_{TRA}	Estimated number of travels between r and r'
c^{TECH}	Hourly technician cost
c^{ENG}	Hourly engineer cost
$c_{rr'}^{\text{TRA}}$	Travel cost from region r to region r'
c^{MW}	Cost of using a maintenance window. This parameter discourages the usage of too many maintenance windows
\bar{n}_r^{TECH}	Number of available technicians during every maintenance window in region r
$\bar{n}_{\text{MAX}}^{\text{CIR}}$	Maximum number of circuits allowed to be migrated per maintenance window
$\bar{n}_{\text{MAX}}^{\text{ENG}}$	Maximum number of engineers available per maintenance window over all regions
$\bar{\Delta}_{ss'}$	Travel time between sites s and s'
$\bar{\Delta}_{\text{MIGR}}$	Time required to migrate one endpoint of a circuit

- $\bar{\Delta}_{\text{SHIFT}}$ Set of possible durations for every shift (e.g., 6 or 8 hours) indexed by δ , of duration $\bar{\Delta}_{\delta}$
- Δ_{δ} duration of the δ^{th} shift in $\bar{\Delta}_{\text{SHIFT}}$.

Variables

Mathematical formulations proposed in Sections 3.3.3 and 3.3.4 are based on the notion of *Technician Shift Configuration* indexed by γ (A configuration is described in detail in Section 3.3.2). Following sets of decision variables are used in these formulations.

- z_{γ} = 1 if configuration γ is selected, 0 otherwise.
- n_{rw} number of technicians used from region r during maintenance window w .
- x_w = 1 if maintenance window w is used, 0 otherwise.

A technician shift configuration is defined in the next section.

3.3.2 Technician Shift Configuration

As mentioned before, the proposed optimization models use the concept of technician shift configuration. Considering all possible configurations, these models select the best combination in order to minimize the cost. A technician shift configuration γ is defined as a set of circuit endpoints that are migrated by one technician located in region r , in maintenance window w . It is characterized by the following parameters:

- V_{γ} Subset of circuit endpoints migrated in configuration γ .
- n_{CIR}^{γ} = $|V_{\gamma}|$, number of circuit endpoints migrated in configuration γ .
- a_v^{γ} = 1 if circuit endpoint v is migrated in configuration γ , 0 otherwise.
- $T_{rr'}^{\gamma}$ = 1 if technician in configuration γ travels to region r' , 0 otherwise.
- $\Delta_{\text{SHIFT}}^{\gamma}$ Duration of technician shift configuration γ i.e., time required for the technician to migrate all circuits endpoints in V_{γ} plus travel times between sites.

Figure 3.2 shows four technician shift configurations, represented by different colors. For example, configuration for technician located in region 1 (in blue) works in sites 1,2 and 3 to migrate four circuit endpoints. This technician works for 40 minutes in total on

migrating circuit endpoints and spends 45 minutes on road in order to travel between sites, thus the duration of the shift for this configuration must be longer than 85 minutes. It is obvious that this shift should occur in a maintenance window which is also at least 85 minutes long. Configuration for red technician located in region 2 includes a travel to region 3. It should be mentioned that travels between regions happen before starting the shifts and the corresponding costs are considered in travel costs.

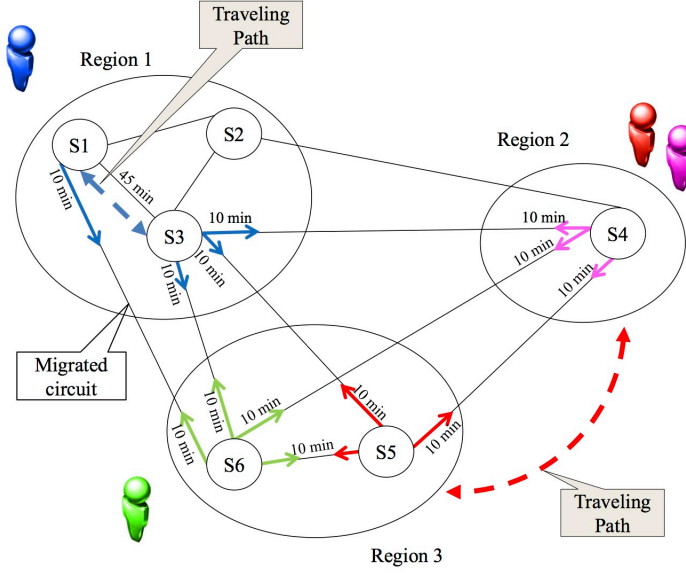


Figure 3.2: Example of four technician shift configurations

Let Γ_{rw} be the set of all possible configurations for a technician located in r during maintenance window w .

We have:

$$\Gamma_w = \bigcup_{r \in R} \Gamma_{rw}, \quad \Gamma_r = \bigcup_{w \in W} \Gamma_{rw},$$

where Γ_w (resp. Γ_r) defines the set of all possible configurations for maintenance window w (resp. region r), and

$$\Gamma = \bigcup_{w \in W} \Gamma_w = \bigcup_{r \in R} \Gamma_r,$$

defines the overall set of possible configurations.

3.3.3 ILP Formulation of STP Phase

In this phase, we take care of the circuit bundles that can keep two technicians busy for the entire duration of a shift. The duration of a full shift for STP is predefined. If we have more than one duration (e.g., 6 and 8 hours), STP is run with respect to each duration consecutively.

Let n^{SHIFT} be the maximum number of circuits with endpoints in the same pair of sites that can be migrated within a single shift. Indicate by $n_{ss'}$, the number of circuits between sites s and s' . For every $s, s' \in S$, we define a partition of $C_{ss'}$ such that:

$$C_{ss'} = \bigcup_{k=1}^{\lfloor \frac{n_{ss'}}{n^{\text{SHIFT}}} \rfloor} C_{ss'}^k \cup C_{ss'}^0,$$

where $C_{ss'}^k \subseteq C_{ss'}$, $k \geq 1$, defines the k^{th} subset of $C_{ss'}$ containing n^{SHIFT} circuits. $C_{ss'}^0$ is the remaining set of circuits i.e., $|C_{ss'}^0| < n^{\text{SHIFT}}$.

An illustration is provided in Figure 3.3. In Figure 3.3(a), there are three subsets of circuits (C_{s_1, s_2} , C_{s_1, s_4} , C_{s_3, s_4}), shown in red, between sites $\{s_1, s_2\}$, $\{s_1, s_4\}$ and $\{s_3, s_4\}$, which can be migrated in the STP phase. To keep the example simple, let us assume that $\lfloor \frac{n_{s_1, s_2}}{n^{\text{SHIFT}}} \rfloor = 1$, so that $C_{s_1, s_2} = C_{s_1, s_2}^1$ with its size equal to n^{SHIFT} , and the same for $\{s_1, s_4\}$ and $\{s_3, s_4\}$. Let $D = \{C_{s_1, s_2}^1, C_{s_1, s_4}^1, C_{s_3, s_4}^1\}$. Every member $d \in D$ defines a complete shift for migration using two technicians, one in each site. Hence, six technicians are required to complete STP in Figure 3.3(a), one in s_2 , one in s_3 , two in s_1 and two in s_4 . Figure 3.3(b) shows the remaining circuits after completing STP in black. Their migration process is discussed in Section 3.3.4.

Note that the partitions of every $C_{ss'}$ are constructed before taking care of technician assignments. Thus the number of required technicians is known in advance, regardless of the technician shift configurations generated in the solution process (Section 3.4.2). The number of required technicians is equal to $2|D|$. Due to this fact, in STP, one can skip the technician and engineer salaries in the migration cost since they represent a fixed cost. Therefore, the goal in STP is to minimize the travel cost of technicians to reach the working

sites (if coming from a neighbor region) and to complete STP in the minimum number of maintenance windows. The objective function is as follows.

$$\min \underbrace{\sum_{r,r' \in R: r \neq r'} \sum_{\gamma \in \Gamma_r} T_{rr'}^\gamma \frac{c_{rr'}^{\text{TRA}}}{\alpha_{\text{TRA}}} z_\gamma}_{\text{travel cost}} + \underbrace{c^{\text{MW}} \sum_{w \in W} x_w}_{\text{\# of maintenance windows}} \quad (3.1)$$

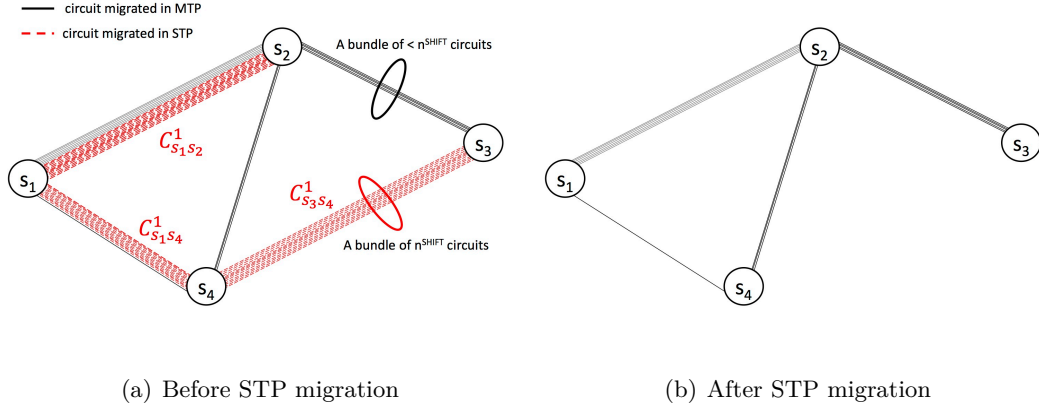


Figure 3.3: Circuits migrated in STP vs. MTP

In the first term of equation (3.1), travel cost, we assume that a technician stays and works in the destination for α_{TRA} consecutive maintenance windows after every travel. For example, if a technician needs to travel from r to r' in 20 configurations and let $\alpha_{\text{TRA}} = 5$, then we assume that the technician will travel 4 times and every time works there for 5 maintenance windows.

In the following, parameters specific to STP are introduced, then its constraints are formulated as linear (in)equalities.

Parameters

Additional set of parameters are required for the formulation of STP phase.

D A set indexed by d , where d is associated with a circuit subset of the type $C_{ss'}^k$, for $k \geq 1$.

$\ell_{sd}^\gamma = 1$ if the technician of configuration γ works in site s to migrate the circuit endpoints related to $d \equiv C_{ss'}^k$, 0 otherwise.

Let $D_{ss'} = \bigcup_{k=1}^{\lfloor \frac{n_{ss'}}{n^{\text{SHIFT}}} \rfloor} C_{ss'}^k \subseteq D$, and $D_s = \bigcup_{s' \in S} D_{ss'}$.

Constraints

There is a number of technical and operational constraints in STP as described below:

- Number of working technicians from region r per maintenance window w cannot be more than the size of the technician pool in this region:

$$\sum_{\gamma \in \Gamma_{rw}} z_\gamma \leq n_{rw} \quad r \in R, w \in W \quad (3.2)$$

$$0 \leq n_{rw} \leq \bar{n}_r^{\text{TECH}} \quad r \in R, w \in W \quad (3.3)$$

- Number of outages per maintenance window is bounded. It amounts to limiting the number of engineers per maintenance window w , i.e., it cannot exceed $\bar{n}_{\text{MAX}}^{\text{ENG}}$:

$$\alpha_{\text{ENG}} \sum_{r \in R} n_{rw} \leq \bar{n}_{\text{MAX}}^{\text{ENG}} \quad w \in W. \quad (3.4)$$

- Total number of migrated circuits in maintenance window w is limited above by $\bar{n}_{\text{MAX}}^{\text{CIR}}$:

$$\sum_{\gamma \in \Gamma_w} n_{\text{CIR}}^\gamma z_\gamma \leq \bar{n}_{\text{MAX}}^{\text{CIR}} \quad w \in W. \quad (3.5)$$

Note that in STP, $n_{\text{CIR}}^\gamma = n^{\text{SHIFT}}$

- If there exists at least one variable n_{rw} that is greater than 0, maintenance window w is used i.e., x_w must be equal to one, whereas if $n_{rw} = 0$ for all regions $r \in R$, then

x_w must be equal to 0:

$$\frac{\sum_{r \in R} n_{rw}}{\sum_{r \in R} n_{rw} + 1} \leq x_w \leq \sum_{r \in R} n_{rw} \quad w \in W. \quad (3.6)$$

- Endpoints located in site s must be processed exactly once:

$$\sum_{\gamma \in \Gamma_{rw}} \ell_{sd}^{\gamma} z_{\gamma} = 1 \quad d \in D_s, s \in S. \quad (3.7)$$

- Endpoints of the circuits in the subsets of $D_{ss'}$ must be migrated in the same maintenance window:

$$\sum_{\gamma \in \Gamma_w} \ell_{sd}^{\gamma} z_{\gamma} = \sum_{\gamma \in \Gamma_w} \ell_{s'd}^{\gamma} z_{\gamma} \quad d \in D_{ss'}, s, s' \in S, w \in W. \quad (3.8)$$

- Domain of the variables are as follows:

$$x_w \in \{0, 1\} \quad w \in W, \quad (3.9)$$

$$z_{\gamma} \in \{0, 1\} \quad \gamma \in \Gamma. \quad (3.10)$$

3.3.4 ILP Formulation of MTP Phase

Unlike the model presented for STP in section 3.3.3, in which the migrated circuit endpoints are pre-determined, in MTP the model decides about the circuit endpoints migrated in the configurations in addition to the technician assignments.

Variables

It is possible that not every circuit can be migrated with the set of available maintenance windows. If this is the case, by introducing variables y_v , we avoid an unnecessary infeasibility of the model, while making the best possible use of the maintenance windows:

$y_v = 1$ if circuit endpoint v is migrated, 0 otherwise.

Objective Function

As stated in Section 4.2.1, the network migration cost consists of salaries along with traveling and maintenance window costs.

The technician salary is hourly pay based upon the number of worked hours during maintenance windows including migration of circuits and/or travels. The engineer is in charge of coordinating a set of α_{ENG} technicians. The engineer's salary is also on an hourly basis.

The objective function of MTP is as follows:

$$\begin{aligned}
 \min \quad & \underbrace{\sum_{\gamma \in \Gamma} c^{\text{TECH}} z_{\gamma} \Delta_{\text{SHIFT}}^{\gamma}}_{\text{technician salary}} + \underbrace{\sum_{w \in W} c^{\text{ENG}} \left(\frac{1}{\alpha_{\text{ENG}}} \sum_{\gamma \in \Gamma_w} z_{\gamma} \Delta_{\text{SHIFT}}^{\gamma} \right)}_{\text{engineer salary}} \\
 & + \underbrace{\sum_{r, r' \in R: r \neq r'} \sum_{\gamma \in \Gamma_r} T_{rr'}^{\gamma} \frac{c_{rr'}^{\text{TRA}}}{\alpha_{\text{TRA}}} z_{\gamma}}_{\text{travel cost}} + \underbrace{c^{\text{MW}} \sum_{w \in W} x_w}_{\text{\# of maintenance windows}} \\
 & - \underbrace{c^{\text{CIR}} \sum_{v \in V} y_v}_{\text{opportunity cost of not migrated circuits}} \tag{3.11}
 \end{aligned}$$

where c^{CIR} is the opportunity cost for one circuit endpoint not migrated in the current planning.

The model aims to provide a plan in order to migrate the maximum number of circuits with minimum cost in minimum number of maintenance windows. In case, there are not enough maintenance windows for the completion of the whole network migration, the fifth term in the objective corresponds to favoring solutions with the maximum number of migrated circuit endpoints.

Constraints

In addition to constraints (3.2)-(3.6) and (3.9)-(3.10), we need two new sets of constraints:

- Every circuit endpoint can be migrated in at most one configuration:

$$\sum_{\gamma \in \Gamma} a_v^\gamma z_\gamma = y_v \quad v \in V. \quad (3.12)$$

- Two endpoints v, v' of circuit $c = \{v, v'\}$ must be migrated during the same maintenance window:

$$\sum_{\gamma \in \Gamma_w} a_v^\gamma z_\gamma = \sum_{\gamma \in \Gamma_w} a_{v'}^\gamma z_\gamma \quad c = \{v, v'\} \in C, w \in W. \quad (3.13)$$

Note that endpoints of a circuit are migrated by two different synchronized technicians and this is guaranteed by the way configurations are generated (See Section 3.4.3).

In the next section, we provide a solution framework for two ILP formulations presented in Sections 3.3.3 and 3.3.4.

3.4 Solution Process

3.4.1 Column Generation and Integer Solutions

The models proposed in Section 3.3 have an exponential number of variables, and therefore are not scalable if solved using classical ILP tools. *Column Generation* (CG) is a technique to manage a solution process that only requires an implicit enumeration of the variables, i.e., the configurations, see, e.g., [28]. Column generation being a decomposition method consists of solving alternatively a restricted master problem (Linear Programming (LP) relaxation of the models presented in sections 3.3.3 and 3.3.4 with a very limited number of columns/variables) and the pricing problem (generation of a new configuration) until the optimality condition is satisfied (i.e., no improving configuration exists anymore). An improving configuration is such that, if added to the current restricted master problem, it will allow a reduction of the current objective value. It is determined by the sign of the reduced cost (negative in a minimization problem). Once the optimal solution of the LP relaxation (z_{LP}^*) has been reached, we will solve exactly the last restricted master problem

using a branch-and-bound method, leading then to an ε -optimal ILP solution (\tilde{z}_{ILP}), where

$$\varepsilon = \frac{\tilde{z}_{\text{ILP}} - z_{\text{LP}}^*}{z_{\text{LP}}^*}.$$

Reader can refer to Chvatal [16] if not familiar with column generation method and the definition of reduced cost. Solution process is summarized in the flowchart represented in Figure 3.4.

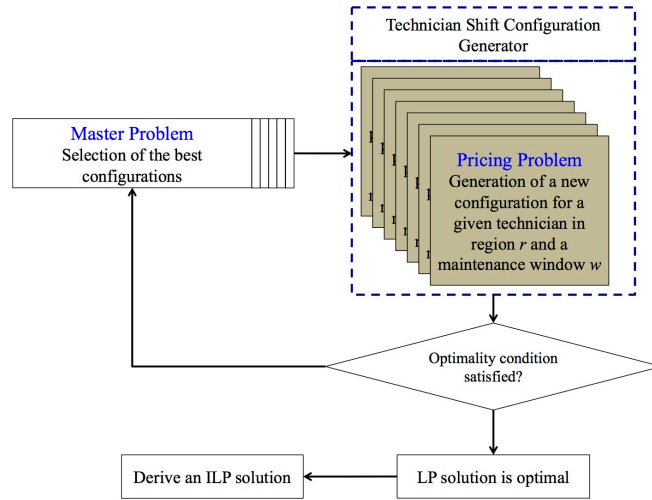


Figure 3.4: CG ILP algorithm

In sections 3.4.2 and 3.4.3, pricing problems of the optimization models STP and MTP are respectively presented. Each pricing problem corresponds to the generation of a configuration, either for the STP or the MTP model. There are as many pricing problems to solve as the number of regions and maintenance windows, as each configuration is defined for a given r and w . The goal in both models is to generate an improving configuration each time we solve a pricing problem. Each pricing problem generates a potential technician configuration in maintenance window w for a technician located in region r . This implies that during a shift, the technician can only take care of circuit endpoints located in r or in neighbor regions $r' \in N_r$ (in which case some travel costs are incurred). Linear programming theory tells us ([16]) that if the optimal value of the pricing problem is negative (as STP and mTP models are minimization problems), then the generated configuration is an

improving one, i.e., is such that, if added to the current restricted master, the optimal value of the resulting restricted master problem will be smaller than the current one.

In order to alleviate the notations, we omit the index γ from the pricing problems.

3.4.2 STP Configuration Generator

Generation of new configurations are delegated to an optimization model called pricing problem. It constructs the most promising configuration, while considering constraints related to one technician working during a maintenance window. In the following, a pricing problem is defined for a region r and maintenance window w .

Reduced Cost

In construction of an improving configuration, we need to find one with a *negative reduced cost*, see [16] if not familiar with linear programming and the concept of reduced cost. A negative reduced cost indicates that introducing the new column (configuration) to the restricted master problem might improve its objective value. In order to maximize the improvement, we search for a configuration with minimum reduced cost. Let $u_{rw}^{(3.2)}$, $u_w^{(3.5)}$, $u_{ds}^{(3.7)}$ and $u_{dss'w}^{(3.8)} \geq 0$ be the corresponding dual values of constraints (3.2), (3.5), (3.7) and (3.8), respectively. The reduced cost of variable z is defined as below:

$$\begin{aligned}
[\text{PP}_{rw}^{\text{STP}}] \quad \min \quad & \sum_{\substack{r' \in R \\ r' \neq r}} T_{rr'} \frac{c_{rr'}^{\text{TRA}}}{\alpha_{\text{TRA}}} - u_{rw}^{(3.2)} - n^{\text{SHIFT}} u_w^{(3.5)} \\
& - \sum_{s \in S} \sum_{d \in D_s} u_{ds}^{(3.7)} \ell_{sd} - \sum_{s \in S} \sum_{s' \in S: s' \neq s'} \sum_{d \in D_{ss'}} (\ell_{sd} - \ell_{s'd}) u_{dss'w}^{(3.8)}. \quad (3.14)
\end{aligned}$$

Constraints

Constraints of $\text{PP}_{rw}^{\text{STP}}$ are as follows.

- In each configuration, the technician can migrate the circuit endpoints in one end of

exactly one $d \equiv C_{ss'}^k$:

$$\sum_{d \in D} \ell_{sd} + \ell_{s'd} = 1 \quad (3.15)$$

- Technician might need to travel to another region:

$$\ell_{sd} \leq T_{rr'} \quad d \in D, r' \neq r, r' = \text{region of } s. \quad (3.16)$$

- At most one travel between regions is possible in each configuration:

$$\sum_{\substack{r' \in R \\ r \neq r'}} T_{rr'} \leq 1. \quad (3.17)$$

- The technician can only travel to regions in the neighborhood of region r :

$$\sum_{r' \notin N_r} T_{rr'} \leq 0. \quad (3.18)$$

- Domain of the variables:

$$\ell_{sd} \in \{0, 1\} \quad d \in D, \quad (3.19)$$

$$T_{rr'} \in \{0, 1\} \quad r' \in R, r' \neq r. \quad (3.20)$$

3.4.3 MTP Configuration Generator

As mentioned in Section 3.3 when defining MTP, each technician can be synchronized with multiple technicians. In other words, during a given shift, a technician may take care of the migration of different circuits, each with one endpoint in his location, and with the other endpoints in different locations, therefore handled by different technicians, see, e.g., the technician in Region 3 in Figure 3.2. Moreover, travel between sites is possible and the duration of shifts can be any number of hours from a predefined set $\bar{\Delta}_{\text{SHIFT}}$. Thus, we need additional sets of variables and different sets of constraints.

Variables

$T_{ss'}$ = 1 if technician needs to travel between sites s and s' , 0 otherwise.

x_δ = 1 if the duration of the configuration (shit) is equal to the δ^{th} duration value in $\bar{\Delta}_{\text{SHIFT}}$, 0 otherwise.

h_s = 1 if technician works in site s , 0 otherwise.

Reduced Cost

Let $u_{rw}^{(3.2)}$, $u_w^{(3.5)}$, $u_v^{(3.12)}$ and $u_{cw}^{(3.13)} \geq 0$ be the dual values of constraints (3.2), (3.5), (3.12) and (3.13). For variable z of MTP formulation, reduced cost is as follows.

$$\begin{aligned} [\text{PP}_{rw}^{\text{MTP}}] \quad \min \quad & (c^{\text{TECH}} + \frac{c^{\text{ENG}}}{\alpha^{\text{ENG}}}) \Delta_{\text{SHIFT}} + \sum_{\substack{r' \in R \\ r \neq r'}} T_{rr'} \frac{c_{rr'}^{\text{TRA}}}{\alpha_{\text{TRA}}} \\ & - u_{rw}^{(3.2)} - n_{\text{CIR}} u_w^{(3.5)} - \sum_{v \in V} u_v^{(3.12)} a_v - \sum_{c=\{v,v'\} \in C} u_{cw}^{(3.13)} (a_v - a_{v'}). \end{aligned}$$

Constraints

Constraints of $\text{PP}_{rw}^{\text{MTP}}$ are as follows.

- Because a technician can only travel to regions in the neighborhood of r , it can only migrate circuit endpoints that belong to these regions:

$$\sum_{v \in \bigcup_{r' \notin N_r} V_{r'}} a_v = 0 \quad (3.21)$$

- Technician can migrate at most one of the endpoints of a circuit:

$$a_v + a_{v'} \leq 1 \quad c = \{v, v'\} \in C'_r, r' \in N_r \quad (3.22)$$

- Duration of a shift is determined by endpoint migration and travel times:

$$\bar{\Delta}_{\text{MIGR}} \sum_{v \in V} a_v + \sum_{s \in S_r} \sum_{\substack{s' \in S_r \\ s' > s}} \bar{\Delta}_{ss'} T_{ss'} \leq \Delta_{\text{SHIFT}} \quad (3.23)$$

- Constraint sets (3.24) and (3.25) together determine the duration of the configuration:

$$\sum_{\delta \in \bar{\Delta}_{\text{SHIFT}}} x_\delta = 1, \quad (3.24)$$

$$\sum_{\delta \in \bar{\Delta}_{\text{SHIFT}}} \bar{\Delta}_\delta x_\delta = \Delta_{\text{SHIFT}}. \quad (3.25)$$

- Technician works in site s if there is at least one endpoint in this site that is processed in the configuration under construction:

$$h_s \leq \sum_{v \in V_s} a_v \leq M h_s \quad s \in S_r, \quad (3.26)$$

where $M \geq 0$ is a big number.

- We need a path going from one site to the next site, if the technician travels between sites. In order to build such path, we add two dummy sites (which do not belong to S), one for each of end of the path, such that the distance from a dummy site to any site is zero. Let $S^+ = S \cup \{s_{\text{SRC}}, s_{\text{DST}}\}$ and $S_r^+ = S_r \cup \{s_{\text{SRC}}, s_{\text{DST}}\}$. Constraints (3.27)-(3.30) together build a valid path.

If the technician works in site s , there should be one site before and one site after s :

$$\sum_{s' \in S_r^+ : s' > s} T_{ss'} + \sum_{s' \in S_r^+ : s' < s} T_{s's} = 2h_s \quad s \in S_r \quad (3.27)$$

There is always exactly one site at the beginning and one site at the end of the path:

$$\sum_{s \in S_r} T_{s_{\text{SRC}}, s} = 1, \quad \sum_{s \in S_r} T_{s, s_{\text{DST}}} = 1. \quad (3.28)$$

Constraint (3.29) ensures that we have a path linking all the visited sites:

$$\sum_{s \in S_r} \sum_{s' \in S_r: s' > s} T_{ss'} = \sum_{s \in S_r} h_s - 1. \quad (3.29)$$

A travel between sites is incurred only if technician needs to handle circuit endpoint migration in both sites s and s' :

$$T_{ss'} \leq \frac{1}{2}(h_s + h_{s'}) \quad s, s' > s \in S_r. \quad (3.30)$$

- A travel to another region is occurred if the technician works in a site that is not located in r :

$$h_s \leq T_{rr'} \quad r' \in N_r : r' \neq r, r' = \text{region of } s. \quad (3.31)$$

- Constraints (3.17) and (3.18) make sure that the technician travels to at most one of the neighbors of region r .
- Domain of the variables

$$a_v \in \{0, 1\} \quad v \in V \quad (3.32)$$

$$T_{ss'} \in \{0, 1\} \quad s, s' \in S \quad (3.33)$$

$$T_{rr'} \in \{0, 1\} \quad r' \in N_r : r' \neq r \quad (3.34)$$

$$h_s \in \{0, 1\} \quad s \in S_r \quad (3.35)$$

$$x_\delta \in \{0, 1\} \quad \delta \in \bar{\Delta}_{\text{SHIFT}} \quad (3.36)$$

$$\Delta_{\text{SHIFT}} \geq 0. \quad (3.37)$$

3.4.4 Overall View of the Solution Process

Telecommunication networks may contain few hundreds to tens of thousands of circuits. Solving a network migration problem with a large number of circuits takes a long time even on high performance machines. Furthermore, companies would also like to have a short term plans for the next few maintenance windows referred to as an *interval* (e.g., 2-3 next maintenance windows) so that they can update the availability of resources including technicians and engineers for the next intervals. We propose a two-phase algorithm (shown in Figure 3.5) that can both handle large instances in reasonable time and provide short term plans. The proposed algorithm returns a set of technician shift configurations called *final shifts* such that all circuits are migrated with respect to technical and operational constraints, with minimum migration cost. In the first phase, the algorithm first identify all eligible circuits to be migrated in STP phase (Section 3.3.3). Generated technician shift configurations from STP are added to final shifts. Availability of technicians and remaining circuits are updated based on the results from STP. If there are still some circuits which are not migrated, MTP is solved for different intervals. At each interval, the algorithm tries to maximize the number of migrated circuits with respect to technician availability and with minimum cost. Then technician availability and remaining circuits are updated again and if there are still some circuits that are not migrated, MTP will be solved for the next interval. Solving MTP is repeated until all circuits are migrated. Using this algorithm gives the companies the opportunity to have a plan for near future maintenance windows, while having the migration plan of the whole network may take a longer time. Another advantage of the proposed algorithm is the ability to handle unexpected changes in availability of resources e.g., technician availability or duration of maintenance windows. In such cases, the technician shifts generated for previous intervals are still valid and the algorithm can take care of circuits in next intervals with respect to updated resources.

Since all circuit endpoints should be migrated, c^{MW} and c^{CIR} in (3.11) need to be defined such that even if one circuit is left after maintenance window w , it will be migrated in maintenance window $w + 1$. Hence, $c^{\text{MW}} - 2c^{\text{CIR}}$ should be negative and $|c^{\text{MW}} - 2c^{\text{CIR}}|$ should

be greater than the cost of migrating two circuit endpoints (i.e., salaries and travel cost).

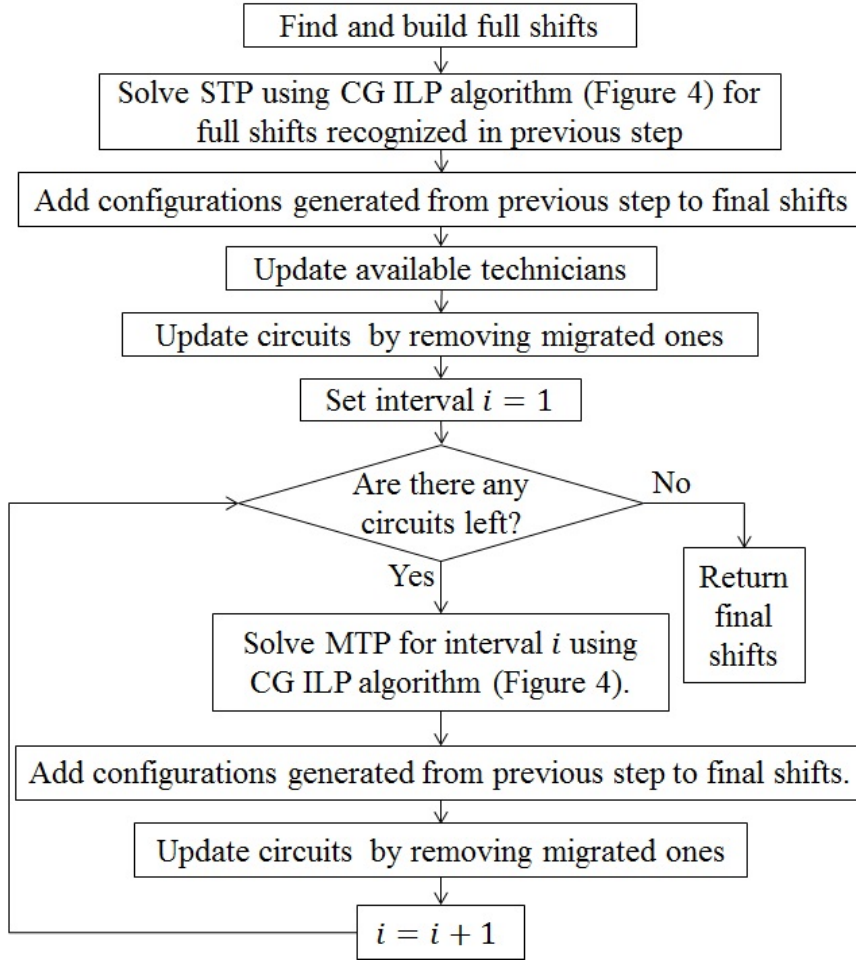


Figure 3.5: Proposed two-phase algorithm

3.5 Numerical Results

In this section, we evaluate the performance of the proposed two-phase algorithm on real networks. Every network migration problem is solved under several scenarios e.g., different values for available technicians, maximum number of engineers ($\bar{n}_{\text{MAX}}^{\text{ENG}}$) and maximum number of circuits ($\bar{n}_{\text{MAX}}^{\text{CIR}}$). Number of available technicians and their distributions over regions are based on different strategies provided by Ciena. Section 3.5.1 describes telecommunication networks studied in this paper.

3.5.1 Data Sets

We have studied four data sets of real networks provided by Ciena customers. These data sets are described in Table 3.1. Figure 3.6 shows the distribution of circuits between pairs of sites in decreasing order. For each data set, number of circuits, sites, regions, average number of circuit endpoints (μ) and their standard deviation (sd) in sites and regions are given. Let DS- $\#_{b,c}^a$ be a scenario for data set DS- $\#$, with $a = \sum_{r \in R} \bar{n}_r^{\text{TECH}}$ (number of available technicians), $b = \bar{n}_{\text{MAX}}^{\text{ENG}}$ and $c = \bar{n}_{\text{MAX}}^{\text{CIR}}$.

Table 3.2 shows the solutions for each data set under different parameters. Each solution contains number of maintenance windows, number of used technicians, average usage of shifts, technician, engineer and travel costs and number of full shifts. Further analysis is given in Sections 3.5.3 through 3.5.6.

Table 3.1: Data sets

Data sets	# of circuits	# of sites	# of regions	Site		Region	
				μ	sd	μ	sd
DS-I	133	43	23	6.18	8.6	11.56	14.44
DS-II	247	20	14	24.70	14.0	35.28	34.70
DS-III	5,824	9	7	1,294.22	690.1	1,664.00	1,001.27
DS-IV	24,268	301	172	161.24	350.84	282.19	519.13

3.5.2 Efficiency of the Solution Process

We first check the efficiency of the solution process. Computational times vary with the number and the characteristics of the set of circuits, i.e., whether phase MTP phase is left with a small or a large number of circuits. Synchronization is easy to take care in the STP phase as all shifts are full with the same node pair, while in the MTP phase, a technician needs to synchronize with at least 2 other technicians.

Computational times vary from 1 minute to around 3 hours for the largest data sets. They include both STP and MTP phases. Since the problem is a static one, indeed a planning problem, we can still talk about a scalable algorithm. In the last column, we

Table 3.2: Migration costs

Data Set	$\sum_{r \in R} \bar{n}_r^{\text{TECH}}$	$\bar{n}_{\text{MAX}}^{\text{ENG}}$	$\bar{n}_{\text{MAX}}^{\text{CHR}}$	# MW	used technicians	average usage of shifts (%)	Cost				Cost per circuit	Full Shifts	
							tech	eng	travel	total		6 hrs	8 hrs
DS-I	35	2	30	6	29	26.71	36,288	9,408	582	46,278	347.95	0	4
			40	5	30	28.94	34,344	8,904	544	43,792	329.26	0	4
		3	30	6	29	28.98	33,912	8,792	537	43,241	325.12	0	4
			40	5	30	30.98	31,752	8,232	503	40,487	304.41	0	4
		4	30	5	28	32.08	31,104	8,064	493	39,661	298.20	0	4
			40	5	28	32.13	30,888	8,008	489	39,385	296.13	0	4
	46	2	30	6	29	26.71	36,288	9,408	575	46,271	347.90	0	4
			40	5	28	28.95	34,128	8,848	540	43,516	327.19	0	4
		3	30	6	29	28.98	33,912	8,792	527	43,231	325.05	0	4
			40	5	30	30.98	31,752	8,232	496	40,480	304.36	0	4
		4	30	5	28	32.13	30,888	8,008	489	39,385	296.13	0	4
			40	5	28	32.50	30,456	7,896	482	38,834	291.99	0	4
DS-II	25	3	30	9	19	30.70	55,512	14,392	423	70,327	284.72	1	5
			40	9	18	33.14	51,624	13,384	393	65,401	264.78	1	3
		4	30	7	18	34.73	49,248	12,768	375	62,391	252.60	1	5
			40	7	19	36.53	46,656	12,096	356	59,108	239.30	1	3
	42	3	30	9	19	31.97	53,784	13,944	412	68,140	275.87	2	4
			40	9	18	34.11	50,328	13,048	386	63,762	258.14	3	4
		4	30	7	18	34.82	49,032	12,712	376	62,120	251.50	3	4
			40	7	19	37.58	45,360	11,760	347	57,467	232.66	3	4
DS-III	28	4	40	231	11	93.57	441,288	114,408	0	555,696	95.41	0	462
			50	122	14	93.89	439,992	114,072	0	554,064	95.13	0	474
		5	40	231	11	93.57	441,288	114,408	0	555,696	95.41	0	462
			50	122	15	94.57	437,616	113,456	0	551,072	94.62	1	472
	63	4	40	231	11	93.57	441,288	114,408	0	555,696	95.41	0	462
			50	122	16	97.39	427,896	110,936	0	538,832	92.52	2	473
		5	40	231	11	93.57	441,288	114,408	0	555,696	95.41	0	462
			50	122	16	97.39	427,896	110,936	0	538,832	92.52	2	473
DS-IV	344	4	85	394	260	68.10	2,395,870	621,152	0	3,017,022	124.32	27	1,631
			100	332	310	68.81	2,375,350	615,832	0	2,991,182	123.26	24	1,628
		5	85	373	258	68.46	2,386,370	618,688	0	3,005,058	123.83	33	1,623
			100	312	313	68.87	2,372,980	615,216	0	2,988,196	123.13	25	1,636
	513	4	85	392	303	68.89	2,372,760	615,160	0	2,987,920	123.12	24	1,636
			100	325	267	69.33	2,359,580	611,744	0	2,971,324	122.44	21	1,639
		5	85	373	263	68.91	2,371,900	614,936	0	2,986,836	123.08	28	1,636
			100	317	313	69.59	2,353,100	610,064	0	2,963,164	122.10	30	1,635

indicate the number of selected configurations in the output solution: it corresponds to the number of required shifts, and hence to the duration of the migration, taking into account the list of maintenance windows.

Observe that data sets DS-II and DS-III have roughly the same number of circuits to migrate in phase MTP, therefore the difference in their computational times comes mainly from phase STP, and slightly in the distribution of their circuits over the network. Computational times for DS-IV are the largest ones, due to the large number of circuits, around 8,000, to be taken care in the MTP phase.

Table 3.3: Efficiency of the algorithm

Data set	# available technicians	$\bar{n}_{\text{MAX}}^{\text{ENG}}$	$\bar{n}_{\text{MAX}}^{\text{CIR}}$	CPU time (min)	# selected tech shifts		
DS I	35	2	30	3	50		
			40	1	50		
		3	30	2	49		
			40	1	45		
		4	30	2	43		
			40	1	41		
	46	2	30	2	50		
			40	3	49		
		3	30	1	46		
			40	1	44		
		4	30	1	41		
			40	1	41		
DS II	25	3	30	16	81		
			40	5	77		
		4	30	9	75		
			40	6	73		
	42	3	30	24	78		
			40	6	76		
		4	30	6	74		
			40	2	68		
		DS III	28	4	40	58	527
					50	43	517
5	40			63	527		
	50			38	512		
63	4		40	61	527		
			50	37	502		
	5		40	59	527		
			50	40	502		
	DS IV		344	4	85	160	4,904
					100	144	4,842
5		85		148	4,884		
		100		108	4,834		
513		4	85	204	4,832		
			100	224	4,792		
		5	85	216	4,830		
			100	146	4,776		

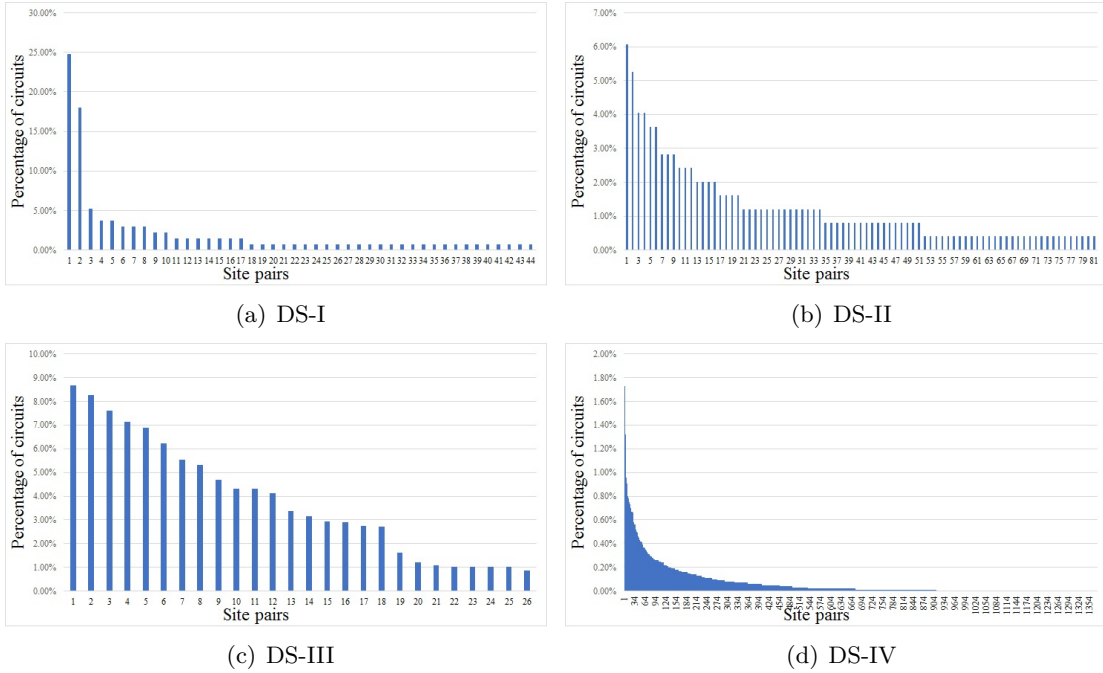


Figure 3.6: Circuit distribution between pairs of sites

3.5.3 Migration Cost Analysis

In Table 3.4, we investigate the average cost per circuit for each data set under different scenarios with respect to the number of available technicians ($\bar{n}_{\text{MAX}}^{\text{ENG}}$) and the limit on the number of migrated circuits per maintenance window ($\bar{n}_{\text{MAX}}^{\text{CIR}}$). Therein, we observe that an increase in either the number of engineers ($\bar{n}_{\text{MAX}}^{\text{ENG}}$) or circuits ($\bar{n}_{\text{MAX}}^{\text{CIR}}$) reduces the cost, except in few cases where the cost remains constant. This is the result of higher shift usage (see Section 3.5.5). Table 3.2 shows that for the same $\bar{n}_{\text{MAX}}^{\text{CIR}}$, the defining factor of shift usage is $\bar{n}_{\text{MAX}}^{\text{ENG}}$. For example in DS-I_{2,30}³⁵ and DS-I_{3,30}³⁵, higher number of $\bar{n}_{\text{MAX}}^{\text{ENG}}$ decreases the cost. It can be explained as follows. If more engineers, and subsequently more technicians, work during a maintenance window, this gives more freedom to synchronize more technicians. In other words, a technician can find more technicians to be synchronized with, which results in more migrated circuits during a shift. This keeps the technicians busy for a larger part of the shift, thus the shift usage improves. From a managerial point of view, one might expect that more resources lead to higher associated costs. However, it is interesting to see that having additional engineers results in lower over cost. This can be explained by the fact

that due to higher shift usage, the number of times a technician and an engineer is used drops.

The above discussion is also true for increasing $\bar{n}_{\text{MAX}}^{\text{CIR}}$. This also gives the algorithm the possibility to use the technicians more efficiently by assigning more circuits to them.

Note that for DS-I, DS-II, and DS-IV, scenarios have different costs while, for DS-III, scenarios DS-III_{4,40}²⁸, DS-III_{5,40}²⁸, DS-III_{4,50}⁶³ and DS-III_{5,50}⁶³ have the same cost. Analysis of circuit endpoint distribution in Table 3.1 sheds some light on these results. DS-III has a higher density of circuit endpoints over sites and regions compared to DS-I, DS-II and DS-IV. It is so dense that the shift usage is already close to its maximum. For this reason, changes in parameters do not show significant impact on the cost in DS-III.

Table 3.4: Average unit circuit cost (\$)

$\bar{n}_{\text{MAX}}^{\text{ENG}}$	$\bar{n}_{\text{MAX}}^{\text{CIRC}}$	# available technicians	Unit circuit cost (\$)	# available technicians	Unit circuit cost (\$)
DS-I					
2	30	35	347.95	46	347.90
3	30		325.12		325.05
4	30		298.20		296.13
2	40		329.26		327.19
3	40		304.41		304.36
4	40		296.13		291.99
DS-II					
3	30	25	284.72	42	275.87
4	30		252.60		251.50
3	40		264.78		239.30
4	40		258.14		232.66
DS-III					
4	40	28	95.41	63	95.41
5	40		95.41		95.41
4	50		95.13		92.52
5	50		94.62		92.52
DS-IV					
4	85	344	124.32	513	123.12
5	85		123.83		123.08
4	100		123.26		122.44
5	100		123.13		122.10

3.5.4 Migrated Circuits in STP vs. MTP

Table 3.5 shows the number of circuits migrated in STP and MTP phases for each data set. DS-II is the only data set with all circuits migrated in MTP. As shown in Figure 3.6, although DS-I has fewer number of circuits than DS-II, its circuits are distributed between 44 site pairs and more than 40% of the circuits are between two pairs of sites. The circuits of DS-I migrated in STP are among the circuits between these sites. In DS-II, the pair with largest percentage is hosting around 6% of the circuits, less than 15 circuits, that is not enough to build a full shift. For this reason, DS-I that is a smaller data set than DS-II has some circuits migrated in STP while all circuits of DS-II are migrated in MTP.

Table 3.5: Number of circuits migrated in STP and MTP

	# total circuits	# circuits migrated in STP	MTP
DS-I	133	48	85
DS-II	247	0	247
DS-III	5,824	5,544	280
DS-IV	24,268	16,176	8,092

Due to the large number of circuits in DS-III and DS-IV, a great volume of circuits are migrated in STP. Since all technician shift configurations generated in STP are associated with full shifts, shift usage for these data sets is higher than DS-I and DS-II.

3.5.5 Shift Usage

Figure 3.7 shows the relation between cost and shift usage. Regardless of the data sets, Figure 3.7 shows that the solutions with lower cost are generated under the parameters resulting in higher shift usage. Note that technicians and engineers are paid for a minimum number of hours or for the full shift (our assumption), even if there is a wait due to synchronization requirements (sometimes called sitting) or if there is no enough circuit migration to occupy them throughout the shift. Indeed, higher shift usage is equivalent to

a more efficient usage of the pool of technicians, but it may be difficult to reach a 100% shift usage due to technician synchronization for each circuit migration.

There may be several reasons why a technician works for a shorter period than that for which he is paid: no other circuits to migrate in his neighbourhood, the number of circuits migrated in the maintenance window is already greater than $\bar{n}_{\text{MAX}}^{\text{CIR}}$, or number of working engineers has reached its maximum of $\bar{n}_{\text{MAX}}^{\text{ENG}}$ i.e., no additional technician can be coordinated by the current engineering team.

For the last two reasons, as explained in Section 3.5.3, increasing $\bar{n}_{\text{MAX}}^{\text{CIR}}$ and/or $\bar{n}_{\text{MAX}}^{\text{ENG}}$ can result in higher shift usage.

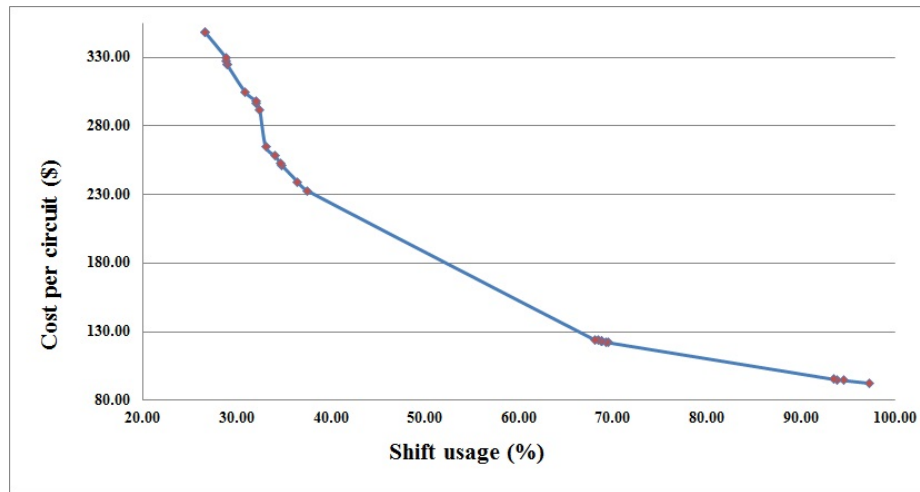


Figure 3.7: Shift usage vs. cost per circuit

3.5.6 Travel cost

As shown in Table 3.2, solutions for data sets DS-III and DS-IV do not contain any travels. The reason should be sought in the nature of these data sets. As Table 3.1 represents, data sets DS-III and DS-IV have high average number of circuits per region. This means that each region has enough number of technicians and they are not required to travel to other regions. However, this is not the case for the first two data sets and travel costs are inevitable in the solutions proposed for them. There are two cases in which travels are inevitable. First, regions with no technicians need one technician for every shift. Second, circuits with both endpoints in the same region, require two technicians to work in that

region over the same maintenance window. If that region does not have the two required technicians, the algorithm has to consider one travel.

Table 3.6 provides travel cost for DS-I and DS-II. As expected, travel cost under scenarios with greater number of available technicians is smaller for both data sets.

Table 3.6: Travel cost (\$)

$\bar{n}_{\text{MAX}}^{\text{ENG}}$	$\bar{n}_{\text{MAX}}^{\text{CIRC}}$	# available technicians	Travel cost (\$)	# available technicians	Travel cost (\$)
DS-I					
2	30	35	582	46	575
3	30		537		527
4	30		493		489
2	40		544		540
3	40		503		496
4	40		489		482
DS-II					
3	30	25	423	42	412
4	30		375		376
3	40		393		386
4	40		356		347

3.6 Conclusions

To the best of our knowledge, this paper is the first try to evaluate time, cost and resource for a network migration problem, defined as the migration of a set of circuits. While the problem was initially solved as a simple heuristic, using the models and algorithms proposed in this paper, it can be solved almost exactly (ε -optimal solution) within few minutes to hours, depending on the number of circuits. The proposed model is a decomposition model and the solution process includes a column generation algorithm. It can provide accurate solutions in terms of the number of required number of technicians, their distribution and the duration of the migration.

Chapter 4

Network Migration Problem: A New Application for Synchronized Vehicle Routing Problem

4.1 Introduction

In this work, we introduce a new application for *vehicle routing problem with synchronized constraints* (VRPS), called *network migration problem* (NMP). Network migration problem arises in telecommunications industries and deals with the process of upgrading the existing infrastructure of a telecommunication network. A telecommunication network (illustrated in Figure 4.1) is represented by a set of demand points (sites) and a set of circuits transmitting traffic between them. Migration of such network is performed by upgrading the circuits one by one. In order to upgrade every circuit, two synchronized technicians simultaneously migrate its two endpoints. The goal of NMP is to find the order of upgrading these circuits such that the associated costs are minimized.

In VRPS, there exists at least one vertex or arc requiring simultaneous visits of vehicles, or successive visits resulting from precedence constraints (a taxonomic review of vehicle

routing problem can be found in [33]). VRPS with simultaneous constraints is categorized into *synchronized arc routing problem* (SARP) and *synchronized node routing problem* (SNRP). In the context of NMP, vehicles and nodes correspond to technicians and sites respectively. In order to migrate every circuit, a technician gets synchronized with another technician such that both technicians simultaneously migrate two endpoints of a circuit located in the sites. In NMP, tasks (upgrading circuit endpoints) are defined in the nodes not on the arcs. Hence, NMP is considered as an SNRP.

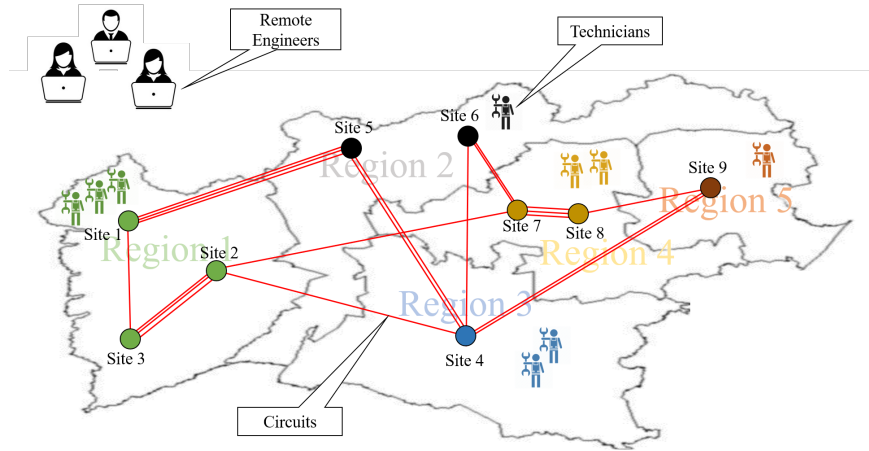


Figure 4.1: An example of a telecommunication network for network migration problem

In SNRP, some nodes may need a service which must be performed by more than one person because the personnel do not have the same expertise. A special case of SNRP is studied in [58], with real applications occurring in home health care systems. A local search heuristic is proposed and the biggest size instance has 45 customers and 16 vehicles. Reinhardt *et al.* [80] investigated the problem of transporting disabled passengers at airports as a dial-a-ride problem such that each passenger might be carried using different transporters and several deliveries and pickups. Here, the synchronization is between different transporters such that the transporter delivering the passenger must meet the transporter picking up the passenger. They proposed a simulated annealing based heuristic. The biggest instance in the paper contains 555 disabled passengers. Bredstrom and Rönnqvist [12] presented a branch and price algorithm for the combined vehicle routing and scheduling problem with synchronization constraints. The problem arises when two

vehicles must meet at a specific point at the same time. The biggest instance includes 80 customers and 16 vehicles, with just 8 customers requiring simultaneous visits.

In SARP, there are tasks to be fulfilled along the arcs and each task is carried out by different synchronized vehicles. An example of an SARP application is *snow plow problem*. Salazar-Aguilar *et al.* [82] studied snow plowing operations, in which every street with more than one lane in each direction should be plowed simultaneously by different synchronized vehicles. They proposed a mixed integer nonlinear formulation, and developed a meta-heuristic method based on adaptive large neighborhood search. The method is evaluated on both artificial and real instances. The biggest artificial instance has 300 intersections and 750 street segments with at most 3 lanes per street and the only real instance includes 430 intersections and 1,056 street segments with at most 2 lanes.

There are also studies considering both SARP and SNRP at the same time. Salazar-Aguilar *et al.* [83] studied road marking operations as synchronized arc and node routing problem such that several capacitated vehicles are used to paint lines on the roads and a tank vehicle is used to replenish the painting vehicles. The generated routes for the vehicles should be synchronized so as to reduce the waiting time at the refill nodes. They proposed an adaptive large neighborhood heuristic. The biggest artificial instance includes 400 nodes and 1,500 arcs. There are also other types of synchronizations in the literature. Interested reader may refer to [30] that classifies VRPS into five categories: task synchronization, operation synchronization, movement synchronization, load synchronization and resource synchronization.

Significance of NMP stems from two sources: the magnitude of the expenses and the impact of the procedure on the efficiency of the network and the customers' satisfaction associated with the migration solution, i.e., the number of disruptions they experience during upgrading the network. It should be mentioned that every circuit migration comes with a disruption as the endpoints are not connected to the equipment. Migration of the network requires several resources, e.g., technicians and engineers. It is considered a strategic decision that can result in massive savings (up to 95% in floor space savings, 90% in power savings [17]). However, the migration might take from several months to a few

years [23] and it is critical to ensure that the process is completed with minimum duration and cost.

The synchronization in the applications of SNRP studied in the literature occurs between the resources that are present in the same node while synchronization in NMP is between two technicians that are not necessarily in the same node as the circuits can be between the same or different sites. While different types of VRPS have been studied in the literature, in order to solve NMP these studies lack the following aspects:

- VRPS has been studied locally with only one depot, meaning that each set of synchronized vehicles belong to the same depot (see, e.g., [11]). However, NMP involves several sets/pools of technicians (i.e., depots) distributed over several regions. Although the regions cannot share technicians, every technician has more choices for the synchronization than single depot VRPS. They can be synchronized with technicians from the same or different regions based on the circuit endpoints and availability of technicians in different depots. Due to the fact that a given technician can be synchronized with several other technicians and every technician synchronization may affect the choices of other synchronizations, it is critical to decide about the optimum choices for every technician.
- From another point of view, the size of NMP depends mainly on number of circuits (tasks) and their distribution over the sites. Telecommunication networks can geographically spread over a continent with hundreds of sites, containing and serving up to thousands of circuits, while the size of the VRPS solved in the literature is far from NMP instances.

While there are different cost models and strategies on suitable time and technology to migrate a telecommunication network (e.g., [75, 95, 2]), there exists only few operational studies for NMP in the literature. Bley *et al.* [10] studied migration of a network as the problem of finding the order of upgrading the links that minimizes the total disruption time. They define the disruption time as the time between upgrading the first link of a path and upgrading the last link, where a path is a set of links. A maximum of K available

technicians are assigned to the sites at each time period t with no travels between sites and no routings. In the first attempt to solve NMP, we proposed a formulation based on migration of each distinct circuit [50]. However, the proposed model was not scalable for large size networks and could be validated for only small instances. In [48], we designed a two-phase algorithm incorporating a column generation model in each phase, in order to provide migration plans for realistic size networks. While the two-phase algorithm provides the solutions that are acceptable by the industry, the quality of the solutions cannot be assessed. None of formulations presented in [50] and [48] provide the exact scheduling for the technicians i.e., both endpoints of a circuit are migrated within a given time limit but not at the same time. Since the solution provided in [50] and [48] is an estimate of NMP solution, it is called a *planning* solution. The solution to NMP that considers the exact scheduling of technicians is called *scheduling* solution. Planning solution is obtained from relaxing some constraints i.e., exact technician synchronization constraints and it provides a lower bound for NMP. This estimate helps network operators have an insight about minimum migration costs. Furthermore, in the solutions provided by the algorithm in [48], swapping the circuits between the same pair of sites does not change the objective value. In other words, this algorithm models NMP in such a way that there exists a high symmetry in the formulation, resulting from the large number of circuits. In this paper, we propose a branch-and-price algorithm based on a new decomposition formulation that provides a planning solution for NMP and does not suffer from the symmetry of the circuits. Since the solution of the proposed branch-and-price algorithm does not consider the exact scheduling for the technicians either, we propose a heuristic that provides the exact scheduling base on the planning solution obtained in branch-and-price algorithm. We also design an algorithm in order to provide a stronger lower bound in the nodes of the proposed branch-and-price method.

The remainder of the paper is as follows. Section 4.2 includes the precise problem statement and complexity of obtaining the optimal planning solution to NMP without considering exact technician scheduling constraints. A new decomposition formulation is presented in Section 4.3. Section 4.4 describes the branch-and-price algorithm and our lower

bound algorithm. In Section 4.5 we propose a heuristic that builds a scheduling solution based on a given planning solution. Section 4.6 includes the numerical experiments on artificial and real instances.

4.2 Problem Statement

In this section, we first provide a formal description of the problem and then discuss the complexity of finding optimal planning solution.

4.2.1 Problem Description

Network migration problem is defined on a telecommunication network represented by a set of *sites* S and a set of *circuits* C between them. The number of circuits between site pairs (s, s') with $s, s' \in S$ is denoted by $\bar{n}_{ss'}$. Every site s is located in a geographical *region* $r \in R$ (e.g., a city). To each region r , \bar{n}_r^{TECH} number of *technicians* is assigned. Regions are not allowed to share technicians and the technicians assigned to a given region r can only work in that region. In addition to two synchronized technicians, every circuit migration requires an *engineer* that works remotely and does not need to be present in the working site. Time required to migrate every circuit $c \in C$ is $\bar{\Delta}_{\text{MIGR}}$. There are at most \bar{n}^{ENG} engineers available, and every engineer can handle up to α^{ENG} technicians. Therefore, it is not always possible to use all available technicians. Migration of the network is performed during a *maintenance window*, which is a period of time usually at night or a specially low traffic time for both telecommunication network and roads. Every maintenance window $w \in W$ has a predefined duration e.g., 8 hours, and all operations have to be completed within this duration. Since migrating every circuit results in a short disruption in the network and the number of disruptions cannot violate clients' Service Level Agreement (SLA) [35], there is a limit \bar{n}^{CIR} on the number of migrated circuits per maintenance window.

A technician working in a given region r during maintenance window w is responsible for a *shift*. A shift is defined as a set of circuit endpoints migrated by a technician in region r during maintenance window w , together with travels between sites, if any. If a shift

belongs to a scheduling solution, it will also include the exact time every circuit endpoint is migrated and called *scheduling* shift. Otherwise, it is called *planning* shift. In this paper, we use shift in order to refer to a planning shift unless it is explicitly mentioned scheduling shift. A given set $\bar{\Delta}_{\text{SHIFT}}$ is used in order to define the possible durations of the shifts, e.g. $\{6h, 8h\}$. Technicians and engineers should be paid for a minimum number of hours per shift. For example, if a technician works for any time less than 6 hours, the payment would be for 6 hours. The migration costs include the payments to the technicians and engineers. Network migration problem is to determine the order of upgrading the circuits in order to minimize the migration costs.

4.2.2 Complexity

In this section we study the complexity of providing optimal planning solution to NMP. The objective of NMP is to minimize the migration costs. In a special case of NMP, when $|\bar{\Delta}_{\text{SHIFT}}| = 1$, minimization of the cost is equivalent to minimizing the total number of the shifts. In the following, we prove that planning the migration of a network for minimizing the number of required shifts is NP-hard. The decision problem is defined as follows:

Given an instance of network migration problem, does m_0 shifts suffice to migrate all the circuits in C ?

Garey and Johnson [40] introduced a basic core of six known NP-complete problems. This core contains *partition problem* which is defined as follows:

Given a set $\{a_1, \dots, a_n\}$, with $a_i \in \mathbb{Z}^+$, $i = 1, \dots, n$, is there a subset $A \subseteq \{1, \dots, n\}$ such that $\sum_{i \in A} a_i = \sum_{i \notin A} a_i$?

It can be proved that partition problem is polynomially reducible to the decision version of NMP with $|\bar{\Delta}_{\text{SHIFT}}| = 1$, and the below theorem follows:

Theorem 4.2.1. *It is NP-complete to decide if an instance of NMP admits a solution with $|S| + 1$ shifts with the same duration.*

Proof. Given a partition problem instance $\{a_1, \dots, a_n\}$, we construct an instance of NMP by setting $S = \{s_1, s_2, \dots, s_{n+1}\}$, $|R| = |S|$, i.e., every site resides in a different region, $|C| = \sum_{i=1}^n a_i$, $\bar{n}_{s_i s_{n+1}} = a_i$ for $i = 1, \dots, n$, and $\bar{n}_{s_i s_j} = 0$ for $j \neq n+1$, $\bar{n}_r^{\text{TECH}} = 1$, $\bar{\Delta}_{\text{SHIFT}} = \{\frac{\sum_{i=1}^n a_i}{2}\}$ and $\bar{\Delta}_{\text{MIGR}} = 1$. It follows that there is a subset $A \subseteq \{1, \dots, n\}$ with $\sum_{i \in A} a_i = \sum_{i \notin A} a_i$ if and only if $|S| + 1$ shifts suffice to migrate all the circuits. \square

In the theorem above, it is proved that every instance of partition problem can be reduced to an instance of network migration problem. In this reduction, we build an instance of NMP with $n + 1$ sites, where for every a_i in partition problem instance there are a_i circuits with one endpoint in site s_i and one endpoint in site s_{n+1} . An example of such reduction is represented in Figure 4.2 for a partition problem instance $\{3, 1, 2, 2, 3, 3\}$. This special instance has no travel between sites, as they are all in different regions. In other words, working hours of the technicians is only the sum of the time spent on migrating circuits, without any travel times. Note that, the minimum time required for migrating all the circuits in site s_{n+1} is $\sum_{i=1}^n a_i$, which is equal to the duration of two planning shifts. With this setup, we need at least 1 planning shift per site $s_i, i = 1, \dots, n$, and at least 2 planning shifts for site s_{n+1} . Hence, if we can do the migration of all the circuits with $|S| + 1$ planning shifts, it means that it is possible to find a subset $A \subseteq \{1, \dots, n\}$ such that $\sum_{i \in A} a_i = \sum_{i \notin A} a_i$. In other words, the technician in site s_{n+1} can spend exactly one shift migrating the circuits in sites $s_i \in \{s_i \in S : i \in A\}$, and migrate the rest of the circuits in another shift.

4.3 Problem Formulation

In this section we propose a new decomposition formulation for NMP. This formulation provides a planning solution and as mentioned before, we use shift instead of planning shift. We apply Dantzig-Wolfe decomposition principles and define a master problem to select the best set of shifts and a set of pricing problems, each responsible for generating shifts for particular regions r and maintenance window w . In this formulation, decision variables of the master problem correspond to shifts. Figure 4.3 represents a subset of possible shifts

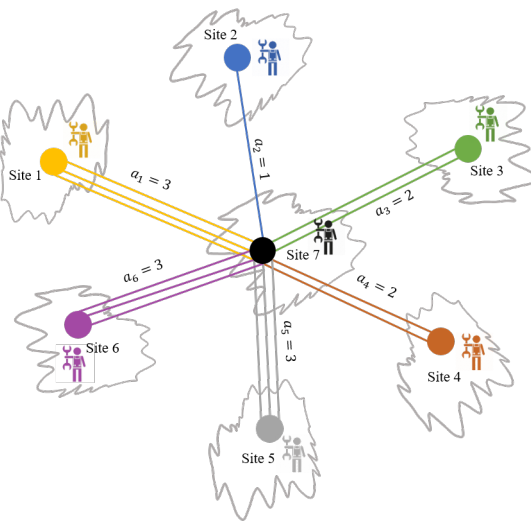


Figure 4.2: An example of reducing an instance of partition problem to an NMP instance as the solution of NMP. The solution considers 3 shifts (for 3 technicians) in region 1, 1 shift in region 2, 1 shift in region 3, 2 shifts in region 4 and 1 shift in region 5. The shift generated for the technician in region 2 includes one travel (dotted line) from site 5 to site 6. Technicians are synchronized by the engineers working remotely. As illustrated in Figure 4.3, technicians might be synchronized with several technicians from the same or different regions.

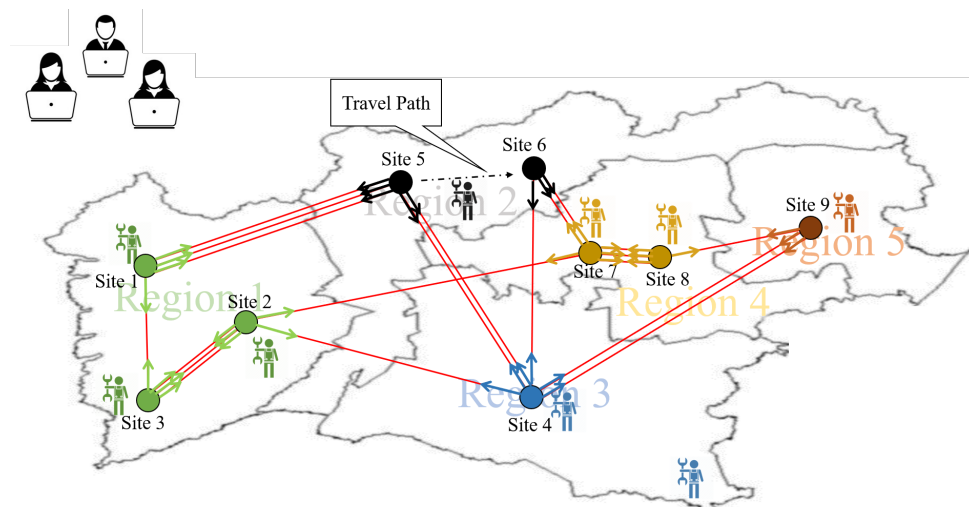


Figure 4.3: A possible solution as a subset of shifts

The remainder of the section is as follows. After introducing the parameters in Section

4.3.1, the problem is formulated in Section 4.3.2, followed by Section 4.3.3 that presents an ILP formulation for generating the shifts.

4.3.1 Parameters

The parameters used in the formulation are listed below:

S	Set of sites (indexed by s).
S	Set of site pairs (s, s') with at least one circuit between them.
R	Set of regions (indexed by r).
S_r	Set of sites in region r .
W	Set of available maintenance windows (indexed by w).
C	Set of circuits (indexed by c).
$C_{ss'}$	Set of circuits between sites s and s' .
$\bar{n}_{ss'}$	Number of circuits between site s and s' ($\bar{n}_{ss'} = \bar{n}_{s's}$).
C_r	Set of circuits with at least one endpoint in the sites $s \in S_r$.
\bar{n}_r^{TECH}	Maximum number of technicians available in region r during every maintenance window.
\bar{n}^{CIR}	Maximum number of circuits allowed to be migrated in a maintenance window.
\bar{n}^{ENG}	Maximum number of engineers available to be used in a maintenance window.
$\text{COST}^{\text{TECH}}$	Hourly cost of a technician.
COST^{ENG}	Hourly cost of an engineer.
$T_{ss'}$	Travel time between sites s and s' .
$\bar{\Delta}_{\text{MIGR}}$	Time required to migrate one endpoint of a circuit.
$\bar{\Delta}_{\text{SHIFT}}$	Set of possible durations for a shift (e.g., 360 or 480 minutes). The set is indexed by δ , with δ th element having duration $\bar{\Delta}_\delta$.
α^{ENG}	The number of technicians supported by one engineer.

4.3.2 Master Problem

As mentioned before, decision variables of the master problem correspond to shifts. Every shift $\gamma \in \Gamma$ is characterized by its duration $\Delta_{\text{SHIFT}}^\gamma$, $n_{ss'}^\gamma$, the number of circuit endpoints migrated from site s connected to site s' and total number of migrated circuit endpoints n_{CIR}^γ . The total set of shifts is denoted by $\Gamma = \bigcup_{w \in W} \Gamma_w = \bigcup_{r \in R, w \in W} \Gamma_{rw}$, where Γ_w is the set of shifts generated for maintenance windows w and Γ_{rw} is the set of shifts to be assigned to a technician located in region r during maintenance windows w . Decision variables $z_\gamma \in \mathbb{Z}^+$ determine the number of times shift γ is assigned to technicians. The master problem is as follows:

$$\min \left(\text{COST}^{\text{TECH}} + \frac{\text{COST}^{\text{ENG}}}{\alpha^{\text{ENG}}} \right) \sum_{\gamma \in \Gamma} \Delta_{\text{SHIFT}}^\gamma z_\gamma \quad (4.1)$$

subject to:

$$\sum_{\gamma \in \Gamma} n_{ss'}^\gamma z_\gamma = \bar{n}_{ss'} \quad s, s' \in S_p \quad (4.2)$$

$$\sum_{\gamma \in \Gamma_w} n_{ss'}^\gamma z_\gamma = \sum_{\gamma \in \Gamma_w} n_{s's}^\gamma z_\gamma \quad s, s' \in S_p, w \in W \quad (4.3)$$

$$\sum_{\gamma \in \Gamma_{rw}} z_\gamma \leq \bar{n}_r^{\text{TECH}} \quad r \in R, w \in W \quad (4.4)$$

$$\frac{1}{\alpha^{\text{ENG}}} \sum_{\gamma \in \Gamma_w} z_\gamma \leq \bar{n}^{\text{ENG}} \quad w \in W \quad (4.5)$$

$$\sum_{\gamma \in \Gamma_w} n_{\text{CIR}}^\gamma z_\gamma \leq 2\bar{n}^{\text{CIR}} \quad w \in W \quad (4.6)$$

$$z_\gamma \in \mathbb{Z}^+ \quad \gamma \in \Gamma \quad (4.7)$$

The objective function (4.1) is equal to NMP costs which is defined as the sum over technician and engineer costs. Constraint set (4.2) assures that all circuits between every two sites s and s' are migrated. Constraint set (4.3) enforces the number of migrated circuits

from site s connected to site s' in maintenance window w to be equal to the number of circuit endpoints migrated from site s' connected to site s . Constraint set (4.4) ensures that no more than \bar{n}_r^{TECH} technicians from region r works in maintenance window w . Constraint set (4.5) controls the number of working engineers during maintenance window w . Constraint set (4.6) guarantees that no more than \bar{n}^{CIR} circuits is migrated in every maintenance window w . Constraint set (4.7) defines the domain of the variables.

Constraint sets (4.2) and (4.3) are too restrictive in finding the optimal ILP solution among the generated columns in column generation method. Assume that we need $m_{ss'}$ circuit between site pair (s, s') to be migrated during maintenance window w in a hypothetical optimal ILP solution. According to constraint sets (4.2) and (4.3), only option is a solution with a set of shifts $\gamma \in \Gamma' \subseteq \Gamma_w$ such that $\sum_{\gamma \in \Gamma'} n_{ss'} z_\gamma = \sum_{\gamma \in \Gamma'} n_{s's} z_\gamma = m_{ss'}$. However, any solution with at least $m_{ss'}$ circuits migrated between site pair (s, s') is feasible. In order to have a higher chance of finding the optimum ILP solution, we use constraint sets (4.8) - (4.11) instead:

$$\sum_{\gamma \in \Gamma_w} n_{ss'}^\gamma z_\gamma = l_{ss'w} \quad s, s' \in S_p, w \in W \quad (4.8)$$

$$m_{ss'w} \leq l_{ss'w} \quad s, s' \in S_p, s < s', w \in W \quad (4.9)$$

$$m_{ss'w} \leq l_{s'sw} \quad s, s' \in S_p, s < s', w \in W \quad (4.10)$$

$$\sum_{w \in W} m_{ss'w} \geq \bar{n}_{ss'} \quad s, s' \in S_p, s < s' \quad (4.11)$$

$$l_{ss'w} \in \mathbb{Z}^+ \quad s, s' \in S_p, w \in W \quad (4.12)$$

$$m_{ss'w} \in \mathbb{Z}^+ \quad s, s' \in S_p, s < s', w \in W. \quad (4.13)$$

where $l_{ss'w}$ is the number of migrated circuit endpoints from site s to site s' in maintenance window w and $m_{ss'w} = \min\{l_{ss'w}, l_{s'sw}\}$. Constraint set (4.8) calculates the number of migrated circuited endpoints from site s to site s' in maintenance window w . Constraint sets (4.9)-(4.10) calculate the number of migrated circuits between sites s and s' in maintenance window w . Constraint set (4.11) assures that all circuit endpoints between sites s and s'

are migrated.

Denote by MF1, the master problem with constraint sets (4.2) - (4.7) and by MF2, the master problem with constraint sets (4.4) - (4.13). It can be proved that by this substitution the optimum value of the LP relaxation remains the same.

Theorem 4.3.1. *Optimum LP value of MF2 ($z_{LP}^*(MF2)$) is equal to the optimum LP value of MF1 ($z_{LP}^*(MF1)$).*

Proof. First, we show that the optimum solution of MF1 is feasible in MF2. Note that in MF2, the solutions are not restricted to migrate the same number of circuit endpoints between s and s' at both ends. Therefore every feasible solution of MF1 including the optimal solution is feasible in MF2. Now we will show that for the optimum solution of MF2, there exists a feasible solution in MF1 with the same migration cost. Consider an optimum solution of the LP relaxation of MF2. If for all $s, s' \in S$ and $w \in W$, $l_{ss'w} = l_{s'sw}$ and $\sum_{w \in W} m_{ss'w} = \bar{n}_{ss'}$, then the solution is also feasible in MF1. Assume that for some site pair $(s, s') \in S$ and some w , $l_{ss'w} \neq l_{s'sw}$. Without loss of generality, let $l_{ss'w} > l_{s'sw}$. In this case, we can remove $l_{ss'w} - l_{s'sw}$ migrated circuit endpoints from site s to site s' in the solution without changing the migration costs. Note that by removing the migration of a circuit endpoint from a shift, the cost either stays the same or decreases. If after removing these circuit endpoints from the solution and making $l_{ss'w} = l_{s'sw}$ the cost decreases, we have a feasible solution for MF1 with a lower optimum than MF2, which is a contradiction since every feasible solution of MF1 is feasible in MF2. Therefore the migration cost stays the same after the deletions. If $\sum_{w \in W} m_{ss'w} = \bar{n}_{ss'}$ after removing the excess circuit endpoints, a feasible solution to MF1 is obtained with the same cost. Otherwise, $\sum_{w \in W} m_{ss'w} - \bar{n}_{ss'}$ circuit endpoint between site pair $(s, s') \in S$ can be removed again and build a feasible solution for MF1. \square

As mentioned before, the advantage of using MF2 is in having higher flexibility in finding the optimal ILP solution among the generated columns in column generation method, however, they both provide the same lower bound (optimum LP value).

4.3.3 The Pricing Problem: An ILP Shift Generator

In this section we propose an ILP formulation that generates a shift γ for a technician located in region r in maintenance window w . For the sake of simplicity in description of this problem we drop the index γ from the decision variables. In addition to $n_{ss'}$ and n_{CIR} described in Section 4.3.2, the following variables are required:

- $h_{ss'}$ = 1 if technician migrates at least one circuit endpoint in site s with the other endpoint in site s' .
- h_s = 1 if technician works in site s in this shift, 0 otherwise.
- $t_{ss'}$ = 1 if a travel from site s to site s' occurs in the shift under construction, 0 otherwise.
- x_δ = 1 if the length of shift under construction is equal to $\bar{\Delta}_\delta$, 0 otherwise.

The pricing problem generating a shift for region r and maintenance window w is as follows:

$$\begin{aligned}
 [\text{PP}_{rw}] \quad \min \quad & \Delta_{\text{SHIFT}} \left(\text{COST}^{\text{TECH}} + \frac{\text{COST}^{\text{ENG}}}{\alpha^{\text{ENG}}} \right) - \sum_{s \in S_r} \sum_{s' \in S} n_{ss'} \pi_{ss'w}^{(4.8)} \\
 & - \pi_{rw}^{(4.4)} - \frac{1}{\alpha^{\text{ENG}}} \pi_w^{(4.5)} - n_{\text{CIR}} \pi_w^{(4.6)}
 \end{aligned} \tag{4.14}$$

subject to:

$$\sum_{s \in S_r} \sum_{s' \in S} n_{ss'} = n_{\text{CIR}} \tag{4.15}$$

$$h_{ss'} + h_{s's} \leq 1 \quad s, s' \in S_r, s \neq s' \tag{4.16}$$

$$h_s \leq \sum_{s' \in S} h_{ss'} \leq M h_s \quad s \in S_r \tag{4.17}$$

$$h_{ss'} \leq n_{ss'} \leq M h_{ss'} \quad s \in S_r, s' \in S \tag{4.18}$$

$$\sum_{s \in S_r} t_{s_{\text{SRC}}, s} = 1, \quad \sum_{s \in S_r} t_{s, s_{\text{DST}}} = 1 \tag{4.19}$$

$$t_{ss'} \leq \frac{1}{2}(h_s + h_{s'}) \quad s, s' > s \in S_r. \quad (4.20)$$

$$\sum_{\substack{s' \in S_r^+ \\ s' > s}} t_{ss'} + \sum_{\substack{s' \in S_r^+ \\ s' < s}} t_{s's} = 2h_s \quad s \in S_r \quad (4.21)$$

$$\sum_{s \in S_r} \sum_{\substack{s' \in S_r \\ s' > s}} t_{ss'} = \sum_{s \in S_r} h_s - 1. \quad (4.22)$$

$$\bar{\Delta}_{\text{MIGR}} \sum_{s \in S_r} \sum_{\substack{s' \in S_r \\ s \neq s'}} n_{ss'} + \sum_{s \in S_r} \sum_{\substack{s' \in S_r \\ s' > s}} T_{ss'} t_{ss'} \leq \Delta_{\text{SHIFT}} \quad (4.23)$$

$$\sum_{\delta \in \bar{\Delta}_{\text{SHIFT}}} x_\delta = 1, \quad (4.24)$$

$$\sum_{\delta \in \bar{\Delta}_{\text{SHIFT}}} \bar{\Delta}_\delta x_\delta = \Delta_{\text{SHIFT}} \quad (4.25)$$

$$n_{\text{CIR}} \in \mathbb{Z}^+ \quad (4.26)$$

$$n_{ss'} \in \mathbb{Z}^+ \quad s, s' \in S_r \quad (4.27)$$

$$h_{ss'} \in \{0, 1\} \quad s, s' \in S_r, s \neq s' \quad (4.28)$$

$$h_s \in \{0, 1\} \quad s \in S_r. \quad (4.29)$$

where $M \geq 0$ is a big number and $S^+ = S \cup \{s_{\text{SRC}}, s_{\text{DST}}\}$ and $S_r^+ = S_r \cup \{s_{\text{SRC}}, s_{\text{DST}}\}$. s_{SRC} and s_{DST} are two dummy sites introduced as the beginning and the end of a path. Objective function (4.14) is the reduced-cost. Constraint (4.15) calculates the number of migrated circuit endpoints in a shift that is equal to the number of circuit endpoints migrated between every site pair (s, s') with $s \in S_r$. Constraint set (4.16) ensures that at most one of the endpoints of every circuit $c \in C_{ss'}$ can be migrated in every shift. Constraint set (4.17) determines the sites where a technician works in the current shift. Constraint set (4.18) assures that all migrated circuit endpoints are from the sites where the technician works in the current shift. Thanks to constraint set (4.19), generated path in this configuration starts form dummy site s_{SRC} and ends in dummy site s_{DST} . Constraint set (4.20) guarantees that if a travel occurs between two sites s and s' , technician will work in both. Constraint set (4.21) determines the sites visited before and after every site $s \in S_r$. Constraint (4.22) ensures that we have a path linking all visited sites. Constraint (4.23) makes sure that the

duration of the shift does not exceed the maximum predefined duration. Constraint sets (4.24) and (4.25) together determine the duration of the shift. Constraint sets (4.26)-(4.29) define the domain of the variables.

4.4 Branch-and-Price Algorithm

In this section, we propose a branch-and-price algorithm in order to obtain an ϵ -optimal ILP solution. After the subproblems do not generate improving columns anymore, the optimal solution of relaxed master problem is obtained. If none-zero variables are all integers, then the optimal ILP solution is equal to the optimal LP solution. Otherwise, some of the variables are fractional and traversing a branch-and-bound tree helps us to reduce the gap between ILP and optimal LP solutions, and find the ϵ -optimal ILP solution. Interested reader may refer to [99, 85] for more information on branch-and-price algorithms.

In order to get an upper bound at every node of the branch-and-bound tree, it is quite common to solve the ILP formulation of the restricted master problem with the set of generated columns. While evaluating the implementation of our branch-and-price algorithm, we observed that solving the ILP formulation of restricted MF2 with an aggregation of generated columns from several nodes considerably improves the quality of ILP solution. This is expected because the more columns is generated, the better chance we have in finding high quality solutions. For this reason, in order to get an ILP solution, we solve MF2 once at every node of the branch-and-price tree with columns generated at that node, and once after traversing a predefined number of nodes using all generated columns so far.

The remainder of this section is as follows. After explaining our branching strategy in Section 4.4.1, we propose an algorithm for improving the lower bound in Section 4.4.2.

4.4.1 Branching Strategy

There are different branching schemes studied in the literature. In addition to perform the branching on the original variables such that the corresponding constraints are added to the subproblems by implementing explicit bounds (see [51, 5]), there are also branching

schemes that focus on the variables of the master problem i.e, branching on the variables corresponding to generated columns or branching on aggregate variables of the original formulation. We are aware that branching on the variables of the master problem corresponding to the generated columns is not efficient and results in an unbalanced tree [97]. This strategy is not consistent with column generation (the issue of regenerating a specified column and looking for k -th best solution of pricing problems if the $(k - 1)$ first solutions are already generated) [99].

In the formulation proposed in Section 4.3.2, circuits and technicians are generic, meaning that they are not indexed and we are just dealing with the *number* of circuits and technicians. In other words, instead of having \bar{n}_r^{TECH} identical subproblems, one for each technician working in region r during a given maintenance window w , there is only one subproblem responsible for generating all shifts. Therefore, decision variables of the pricing problems are not suitable for branching (see [100]).

Branching on the aggregated variables of the original formulations, in case of identical subproblems, is not typically sufficient to eliminate all fractional solutions. Vanderbeck [99] proposes a generic branching scheme based on the aggregated value of original variables. Since this branching does not suffice the integrality of the solution, once the integrality of original variables is achieved, the proposed scheme is applied to remove fractional solution. The basis of the proposed scheme in [99] is to return to non-identical subsystems by inducing symmetry via introducing new subproblems dynamically. The branching constraints can then be enforced in the pricing problems. They show that in a well structured IP whose constraint matrix has R diagonal blocks, at most R subproblems will be introduced but with a significant increase in the number of calls to the pricing problems. In terms of NMP, we are facing $\sum_{w \in W} \sum_{r \in R} \bar{n}_r^{\text{TECH}}$ subproblems if we decide to induce symmetry that is a big number. Furthermore, due to existing symmetry between maintenance windows in the proposed NMP formulation i.e., swapping the shifts defined in two different maintenance windows results in two solutions that have the same LP value with different variable values, even the integrality of the aggregated value of original variables is not easily achieved specially in large NMP instances. The reason is that when a fractional solution is removed

at a node of the tree, it shows up again with different variable values somewhere else in the tree. In this paper, we do the branching based on the value of aggregated variables $n_{ss'}$. This branching scheme was first tested in [98]. Although this branching scheme theoretically does not guarantee the integrality of the solution, it experimentally returns the integral solution for some instances. It should be mentioned that the obtained solution may still be fractional even if no more branching constraint can be found. The same approach is also used in [99] to compare with the proposed generic scheme.

In our branch-and-price algorithm, we branch on the number of migrated circuits between two sites s and s' in maintenance window w . Following Constraint sets (4.9) - (4.11), decision variable $m_{ss'w}$ in MF2 calculates this number. The set of $\{\bar{s}, \bar{s}', \bar{w}\}$ for branching is selected as the most fractional $\hat{m}_{ss'w}$, the solution obtained from the LP relaxation of MF2:

$$\{\bar{s}, \bar{s}', \bar{w}\} = \arg \min_{s, s', w} \{ |\hat{m}_{ss'w} - \lfloor \hat{m}_{ss'w} \rfloor - 0.5| \}, \quad (4.30)$$

Equation (4.30) defines the site pair, maintenance window and the right hand side of the new branch. Let $\hat{m}_{\bar{s}\bar{s}'\bar{w}}$ be the most fractional solution. Then, the new branches are defined as follows:

$$m_{ss'w} \leq \lfloor \hat{m}_{\bar{s}\bar{s}'\bar{w}} \rfloor, \quad m_{ss'w} \geq \lfloor \hat{m}_{\bar{s}\bar{s}'\bar{w}} \rfloor + 1. \quad (4.31)$$

4.4.2 Strengthening the Lower Bound

In this section, we propose an alternative formulation in order to obtain a stronger lower bound during branch-and-price algorithm. In the new column generation formulation, every column $p \in P$ represents a plan for a given maintenance window w , i.e., a set of shifts corresponding to a set of technicians working during maintenance window w . $P = \bigcup_{w \in W} P_w$ is the set of all plans. A plan p is characterized by $m_{ss'}^p$, the number of migrated circuit endpoints from s to s' , and $cost^p$, sum of the migration costs of its shifts. Decision variable v_p is equal to 1 if plan p is selected and 0 otherwise. The master problem AltMP is as

follows:

$$[\text{AltMP}] \quad \min \sum_{p \in P} \text{cost}^p v_p \quad (4.32)$$

subject to:

$$\sum_{p \in P_w} v_p \leq 1 \quad w \in W \quad (4.33)$$

$$\sum_{p \in P} m_{ss'}^p v_p \geq \bar{n}_{ss'} \quad s, s' \in S_p \quad (4.34)$$

$$v_p \in \{0, 1\} \quad \gamma \in \Gamma. \quad (4.35)$$

Objective function (4.32) calculates the migration cost defined as in Section 4.2.1. Constraint set (4.33) assures that at most one plan is selected for every maintenance window w . Constraint set (4.34) guarantees that all circuit endpoints are migrated. Constraint set (4.35) defines the domain of the variables.

In this formulation, the pricing problem generates a plan for a given maintenance window w . The cost of a plan is equal to the sum over the cost of the shifts assigned to the technicians. In order to have a plan p for maintenance window w , the model needs to decide which technicians are going to work during the maintenance window and assign a shift to each working technician. In other words, unlike model (4.14) - (4.29), the pricing problem needs to differentiate between technicians, hence the technicians have their own indices. Denote by GPP_w , the pricing problem generating a generic plan for maintenance window w . Since the size of GPP can be very large and solving such an ILP is very costly, we consider another pricing problem PrPP that takes advantage of the shifts so far generated in the nodes of branch-and-bound tree and generates plans using the set of precomputed shifts (Figure 4.4). Once no improving plan, i.e., a plan with negative reduced cost is generated from the set of precomputed shifts, GPP_w is solved one time for every w . If it detects that LP objective of (4.32) can still be improved, we stop and continue with branching.

Otherwise, optimum LP solution of AltMP is obtained and it cannot improve the lower bound further. See .A and .B for more details on the formulations of these two pricing problems.

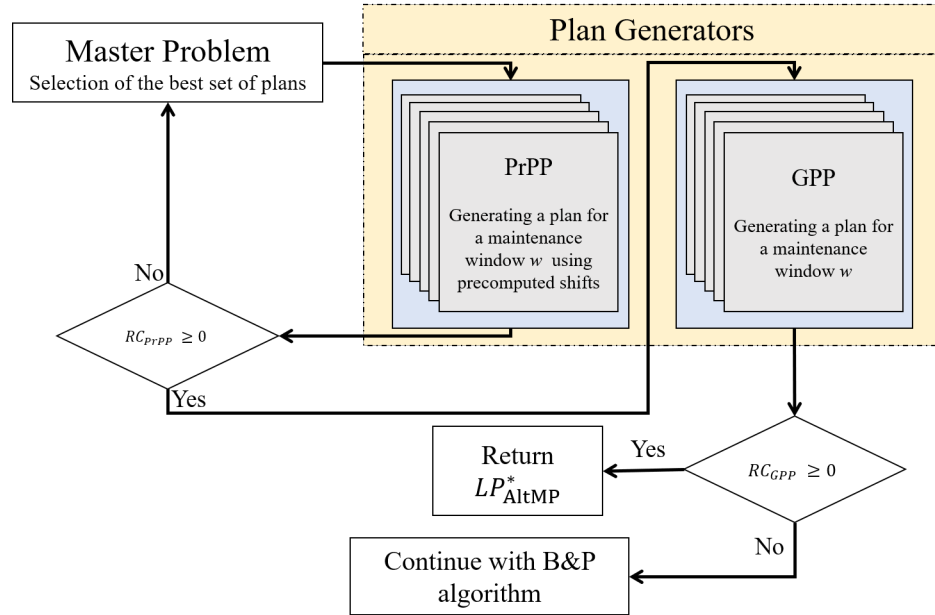


Figure 4.4: Algorithm for strengthening the lower bound

After traversing a predefined number of nodes in the branch-and-bound tree, we run the algorithm presented in Figure 4.4. If the returned value is optimum i.e., GPP_w cannot generate an improving plans for any w , the lower bound is updated and we do not run the algorithm again. Otherwise, we keep branching in the tree and consider running the algorithm again after traversing the predefined number of nodes.

Although the algorithm presented in Figure 4.4 is designed to strengthen the lower bound, AltMP is not suitable to be used instead of MF2 in the nodes of branch-and-price algorithm. The reason is that GPP is too large to be solved multiple times in the column generation method, while it is now solved just once in order to check the optimality of the solution obtained by using PrPP.

4.5 Heuristic for Deriving a Scheduling Solution from a Planning Solution

The formulation presented in Section 4.3.2 does not provide an exact scheduling solution for NMP as the scheduling constraints are missing. In the proposed formulation, both endpoints of every circuit are forced to be migrated in the same maintenance window but not at the same time. It is possible to need more shifts that are also longer while considering the exact scheduling due to the waiting and idle times. We propose Algorithm 1 to provide an exact scheduling form the planning solution.

Algorithm 1 Scheduling Solution Builder

```

1: Input: A planning solution i.e., a set of planning shifts  $\Gamma^p$ 
2: Output: A scheduling solution i.e., a set of scheduling shifts  $\Gamma^s$ 
3: Set  $\Gamma^s \leftarrow \emptyset$ .
4: Set the duration of every shift  $\gamma \in \Gamma^p$  to maximum value  $(\Delta_{|\Delta|})$ .
5: for  $w \in W$  do
6:   Sort the shifts in  $\Gamma_w^p$  based on the shift usage decreasingly.
7:   Build an empty shift temp_shift $_\gamma$  for every  $\gamma \in \Gamma_w^p$ .
8:   Set completion_time $_\gamma = 0$  for every  $\gamma \in \Gamma_w^p$ .
9:   for  $\gamma \in \Gamma_w^p$  do
10:    for every endpoint  $e$  of circuit  $c$  migrated in shift  $\gamma$  do
11:      if  $t_c$  is already set then
12:        continue.
13:      Find the shift  $\gamma'$  responsible for the other endpoint  $e'$  of circuit  $c$ .
14:      Find the earliest time that circuit  $c$  can be migrated ( $t_c$ ).
15:      if  $t_c \leq \Delta_{|\Delta|}$  then
16:        Set  $t_c$  for both endpoints  $e$  and  $e'$ .
17:        completion_time $_\gamma = \max\{t_c, \text{completion\_time}_\gamma\}$ .
18:        completion_time $_{\gamma'} = \max\{t_c, \text{completion\_time}_{\gamma'}\}$ .
19:      else
20:        Remove circuit endpoint  $e$  from shift  $\gamma$  and add to temp_shift $_\gamma$ .
21:        Remove circuit endpoint  $e'$  from shift  $\gamma'$  and add to temp_shift $_{\gamma'}$ .
22:      Update the duration of shift  $\gamma$  based on completion_time $_\gamma$ .
23:       $\Gamma^s = \Gamma^s \cup \gamma$ .
24:   Set  $\Gamma_{|W|+1}^p \leftarrow \emptyset$ .
25:   for  $\gamma \in \Gamma_w^p$  do
26:     if temp_shift $_\gamma$  is not empty then
27:       Add temp_shift $_\gamma$  to  $\Gamma_{|W|+1}^p$ 
28:   if  $\Gamma_{|W|+1}^p$  is not empty then
29:      $W = W \cup \{|W| + 1\}$ .
30:      $\Gamma^p = \Gamma^p \cup \Gamma_{|W|+1}^p$ 
31: Return  $\Gamma^s$ .

```

Algorithm 1 starts with the set of planning shifts obtained from the proposed branch-and-price algorithm (Γ^p) in Section 4.4 and returns a set of scheduling shifts (Γ^s). The duration of every shift $\gamma \in \Gamma^p$ is set to the maximum possible duration ($\Delta_{|\Delta|}$). For every maintenance window $w \in W$, the planning shifts are sorted decreasingly based on the shift

usage that indicates the portion of the shifts technicians are actually working i.e., migrating the circuits or traveling between sites. For every planning shift $\gamma \in \Gamma^p$ in maintenance window w , a temporary planning shift (temp_shift_γ) is built. The temporary shifts are initially empty. We set the exact migration time (t_c) for the endpoint of the circuits in the busiest shifts and the migration time of the other endpoints are determined accordingly. The exact migration time t_c is the sum of the migration times for the endpoints migrated before, possible travel times that is the time required to travel between two sites and the migration time of the current circuit endpoint. If two consecutive migrated endpoints are in the same site, the travel time is equal to 0. When migrating a circuit requires the shifts to become longer than $\Delta_{|\Delta|}$, both endpoints are added to the corresponding temporary shifts and removed from the original planning shifts. Once the exact migration times for the endpoints migrated in a shift $\gamma \in \Gamma_w^p$ are all set, the duration of shift γ is modified based on the exact migration time of the last migrated circuit endpoint ($\text{completion_time}_\gamma$). The scheduling shift is then added to Γ^s . After going through every planning shift $\gamma \in \Gamma_w^p$, if there are some temporary shifts that are not empty, the algorithm will consider a new maintenance window $|W| + 1$ and add it to the set of maintenance windows $W = W \cup \{|W| + 1\}$. The temporary shifts will be assigned to maintenance window $|W| + 1$.

4.6 Numerical Results

In this section we have provided the numerical experiments to evaluate the branch-and-price algorithm presented in Section 4.4. In Section 4.6.1, the data sets used in the evaluation are introduced. Section 4.6.2 examines the performance of the branch-and-price algorithm in terms of solution quality and cpu time. Section 4.6.3 conducts cost analysis on the solutions of NMP instances.

All numerical experiments have been obtained with running the programs on a server with Intel(R) Xeon(R) CPU E5-2687W v3 at 3.10GHz and a memory limit of 8GB. We employed CPLEX V12.8.0 for solving LP and ILP formulations.

4.6.1 Data sets

We utilized two sets of instances in our numerical experiments. Instances of these data sets vary in size from very small to big networks, with different circuit endpoint distributions in order to cover all possible cases. The first set consists of 10 artificial instances showing the efficiency of the algorithm in extreme cases. In the process of generating challenging data instances, we observed that the gap between LP and ILP solutions has correlation with the notion of *bundles*. A bundle is a particular number of circuits between two sites that can keep two technicians busy for exactly one shift. As shown in Table 4.1, instances are characterized by the number of circuits, sites, regions, mean and standard deviation of the number of circuit endpoints over sites and regions, number of bundles and the percentage of circuits in bundles. These artificial instances represent extreme circuit endpoint distributions such that in the first five instances there is no bundle while in the last half of the data sets, all circuits are in bundles. In telecommunication networks, as the number of circuits grows, it is more likely to have bundles. As mentioned previously, every circuit transmits data between two endpoints. Each endpoint is a component of the network deployed in a site. Since both the space and components are expensive, tens to hundreds of components are deployed in every site and each equipment can handle hundreds of circuits. Network operators try to exploit maximum capacity that results in higher number of bundles in instances with large number of circuits.

Table 4.1: Artificial data set

Data instance	C	S	R	Endpoint distribution over				#	% circuits in bundles
				sites		regions			
				μ	sd	μ	sd	bundles	
1	50	10	8	10.00	8.88	12.50	13.37	0	0
2	100	12	10	14.28	11.49	20.00	15.76	0	0
3	300	15	9	40.00	24.77	66.67	57.99	0	0
4	500	15	12	62.67	32.14	83.33	52.16	0	0
5	750	15	14	100.00	41.65	107.14	42.82	0	0
6	1,008	20	12	100.80	56.08	168.00	98.47	42	100
7	1,320	20	15	132.00	69.35	176.00	106.20	55	100
8	1,512	20	14	151.20	48.65	216.00	120.16	63	100
9	2,016	23	16	173.30	54.56	252.00	180.19	84	100
10	2,520	23	17	219.13	70.87	296.47	135.24	105	100

The second data set includes 22 instances provided by customers of Ciena and are called

real instances in this paper. Table 4.2 describes these data instances. In this data set, every instance corresponds to a real network that is considered for migration. As shown in Table 4.2, number of circuits in bundles in big networks increases.

Table 4.2: Real data set

Data instance	C	S	R	Endpoint distribution over				# full	% circuits in bundles
				sites		regions			
				μ	sd	μ	sd	bundles	bundles
DS-1	133	43	23	6.18	8.60	11.56	14.44	2	36.09
DS-2	185	23	9	16.08	4.34	41.11	21.20	0	0.00
DS-3	247	20	14	24.70	14.0	35.28	34.70	0	0.00
DS-4	300	23	11	26.08	14.06	54.54	29.56	2	16.00
DS-5	350	23	9	30.43	18.02	77.78	43.48	4	27.42
DS-6	450	32	14	28.12	12.19	64.28	32.88	5	26.66
DS-7	550	29	13	37.93	26.05	84.61	78.33	10	43.63
DS-8	650	33	13	39.39	31.92	100.00	97.39	10	36.92
DS-9	750	37	15	40.54	28.37	100.00	74.64	12	38.40
DS-10	850	47	19	36.17	34.04	89.47	104.82	14	39.52
DS-11	950	42	14	45.23	31.83	135.71	71.71	21	53.05
DS-12	1,000	54	29	37.03	27.20	68.96	78.61	17	40.80
DS-13	1,500	72	41	41.66	35.45	73.17	81.22	21	33.60
DS-14	2,000	65	23	61.53	65.85	73.91	194.53	52	62.40
DS-15	2,500	72	25	69.44	79.63	200.00	243.33	68	65.28
DS-16	3,000	72	23	83.33	88.04	260.80	229.71	87	69.60
DS-17	3,500	114	90	61.40	65.79	77.77	79.89	66	45.25
DS-18	4,000	109	83	73.39	90.34	96.38	117.69	107	64.20
DS-19	4,500	36	11	250.00	277.50	818.88	797.74	174	92.80
DS-20	5,000	87	30	114.94	137.27	333.33	415.54	164	78.72
DS-21	5,500	60	17	183.33	196.24	647.05	576.05	200	87.27
DS-22	5,824	9	7	1,294.22	690.10	1,664.00	1,001.27	231	95.19

4.6.2 Performance and Efficiency of the Solution Method

Tables 4.3 and 4.4 present the results of solving artificial and real instances respectively. Every instance is solved with at least one value for \bar{n}^{CIR} and \bar{n}^{ENG} . We set a time limit of 300 seconds for solving the instances.

As shown in Tables 4.3 and 4.4, all artificial and real instances are solved with small gap (less than 10%) within the time limit. Artificial instances with all circuits in bundles (instances 6-10), despite being very large, are solved very fast (less than 2 minutes) with a small gap (less than 4%). The artificial instances with no circuits in bundles (instances 1-5), come with bigger gaps although they have smaller size. This is also true for real instances and the biggest instance with most of the circuits in bundles has the smallest gap (1.30%).

A comparison between columns LP_{root} , $\text{LP}_{\text{AltMP}}^*$ and $\overline{\text{ILP}}$ in Tables 4.3 and 4.4 shows that the LP solution of MF2 is a relatively strong lower bound for larger instances, with a

gap less than 12% when comparing against $\overline{\text{ILP}}$, for DS-8 to DS-22. The LP relaxation of AltMP is relatively strong for all data instances, with a gap less than 10% from $\overline{\text{ILP}}$. This shows the effectiveness of incorporating AltMP in our branch-and-price algorithm, which despite being harder to solve, helps to better assess the quality of the obtained ILP solution. On the other hand, column ILP_{root} shows that, if any ILP solution is found in the root node, it is usually far from the value of $\overline{\text{ILP}}$ at the end of the algorithm, which proves that it is worth the effort to employ branch-and-price algorithm in order to find better ILP solutions.

Table 4.3: Algorithm performance on artificial data set

Data instance	\bar{n}^{CIR}	\bar{n}^{ENG}	LP_{root}	ILP_{root}	$\text{LP}_{\text{AltMP}}^*$	$\overline{\text{ILP}}$	Gap (%)	cpu time (S)
1	25	1	8,112.28	14,683.68	8,819.53	9,789.12	9.90	300
2	25	1	12,689.60	21,481.67	15,051.57	16,587.12	9.26	300
3	35	1	28,098.40	44,866.80	28,785.34	32,018.08	10.10	300
4	50	2	45,455.96	64,988.88	45,646.30	50,430.80	9.49	300
5	75	2	67,980.00	84,567.11	67,980.00	75,865.67	10.39	300
6	90	2	91,365.12	91,365.12	91,365.12	91,365.12	0.00	45
7	90	2	119,644.79	123,995.52	119,644.79	123,995.52	3.50	35
8	90	2	137,047.68	137,047.68	137,047.68	137,047.68	0.00	72
9	100	3	182,730.24	182,730.24	182,730.23	182,730.23	0.00	68
10	100	3	228,412.79	234,938.88	228,412.79	234,938.88	2.77	83

A comparison between the values in column $\text{LP}_{\text{AltMP}}^*$ for different instances in both Tables 4.3 and 4.4 demonstrate that the algorithm for strengthening the lower bound results in higher improvement for smaller data instances with a low ratio of the circuits in bundles. The initial lower bound for the instances with high number of bundles is already strong enough and the $\overline{\text{ILP}}$ solution is within an acceptable gap from the initial LP.

As mentioned before, the time limit for solving the instances is 300 seconds. In order to evaluate the impact of this time limit, real instances are solved considering two time limits: 300 and 900 seconds. Surprisingly, the solution found within 300 seconds is the same as the solution found in 900 seconds and we observed no improvement after 300 seconds.

Table 4.4: Algorithm performance on real data set under 300 seconds time limit

Data instance	\bar{n}^{ENG}	\bar{n}^{CIR}	LP_{root}	ILP_{root}	LP_{AltMP}^*	\overline{ILP}	Gap (%)
DS-1	2	25	26,557.52	80,940.30	29,540.55	31,568.70	6.42
		30	26,557.52	79,400.00	28,864.23	29,991.20	3.76
DS-2	2	25	24,885.00	88,955.80	28,900.40	31,833.90	9.22
		30	24,885.00	85,654.79	27,650.00	29,367.36	5.85
DS-3	2	25	28,131.27	-	32,682.28	34,945.00	6.48
		30	28,131.27	181,098.72	31,414.09	33,718.08	6.83
DS-4	2	25	27,288.56	95,509.96	28,460.32	30,570.45	6.90
		30	27,288.56	-	28,098.40	29,911.20	6.06
DS-5	2	35	29,379.20	-	31,825.36	35,200.73	9.59
		45	29,379.20	35,893.43	31,724.00	34,805.76	8.85
DS-6	2	35	36,709.20	-	42,540.82	46,772.20	9.05
		45	36,709.20	45,682.56	40,788.00	45,138.72	9.64
DS-7	2	35	46,881.60	-	50,135.46	55,432.80	9.56
		45	46,881.60	55,743.60	49,852.00	54,927.84	9.24
DS-8	2	35	59,097.28	-	59,450.30	65,147.48	8.75
		45	59,097.28	66,620.40	59,097.28	64,716.96	8.68
DS-9	2	35	68,438.50	-	68,438.50	75,327.25	9.15
		45	67,980.00	-	67,980.00	74,506.08	8.76
DS-10	2	35	77,441.00	-	78,352.34	83,998.35	6.72
		45	74,625.12	-	77,361.24	83,479.44	7.33
DS-11	2	45	84,660.35	-	86,289.28	94,900.07	9.07
DS-12	2	45	87,902.30	152,241.91	91,002.56	98,435.04	7.55
DS-13	3	60	131,949.28	174,722.47	136,186.60	149,555.99	8.94
DS-14	3	60	181,415.96	-	181,415.96	194,694.71	6.82
DS-15	3	60	226,871.92	286,359.51	226,871.92	244,728.00	7.30
DS-16	3	80	272,191.92	296,664.71	272,191.92	289,322.87	5.92
DS-17	3	80	317,511.92	368,995.44	317,511.92	349,689.11	9.20
DS-18	3	80	362741.28	-	362741.28	386942.16	6.25
DS-19	4	100	407,880.00	-	407,880.00	423,651.36	3.72
DS-20	4	100	453,245.32	-	453,245.32	480,754.56	5.72
DS-21	4	100	498,565.32	-	498,565.32	522,902.15	4.65
DS-22	4	100	527,887.36	551,453.75	527,887.36	534,866.64	1.30

4.6.3 Network Migration Cost Analysis

Table 4.5 presents the cost analysis of real instances. Columns 4-7 are the components of the planning solution returned by branch-and-price algorithm: the number of used maintenance windows, number of planned shifts, the cost of migrating every circuit and the shift usage. Columns 8-10 show contain the same information as columns 4-7 in the scheduling solution provided by Algorithm 1. The last column show the cost difference between planning and scheduling solutions.

Figure 4.5 shows that as the size of the networks grows, the shift usage increases and migrating every circuit becomes cheaper. This happens because the technicians are paid a minimum number of hours per shift even if only one circuit endpoint is migrated. Hence, the model tries to maximize the number of migrated circuit endpoints once a shift is assigned

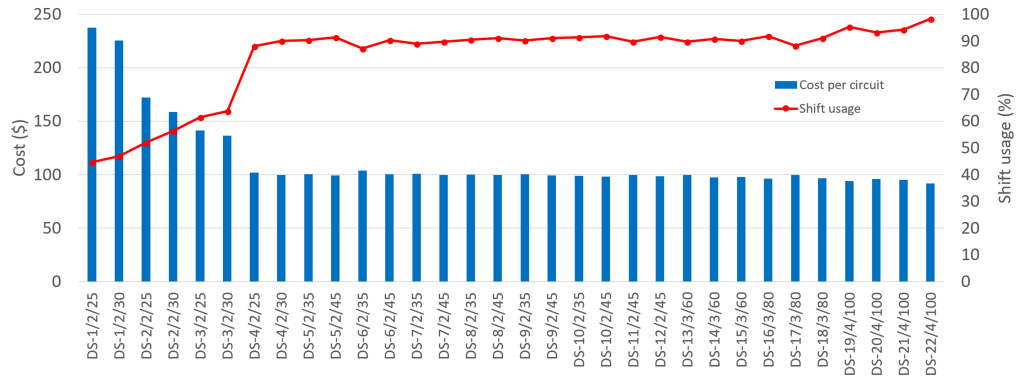


Figure 4.5: Migration cost vs. shift usage

to a technician. The difficulty in smaller networks is that there are not enough number of circuit endpoints to make the most efficient use of a shift. As illustrated in Figure 4.5, the cost spent for every circuit reduces as the shift usage increases, which is expected. This means that more circuits are migrated during the shifts on average.

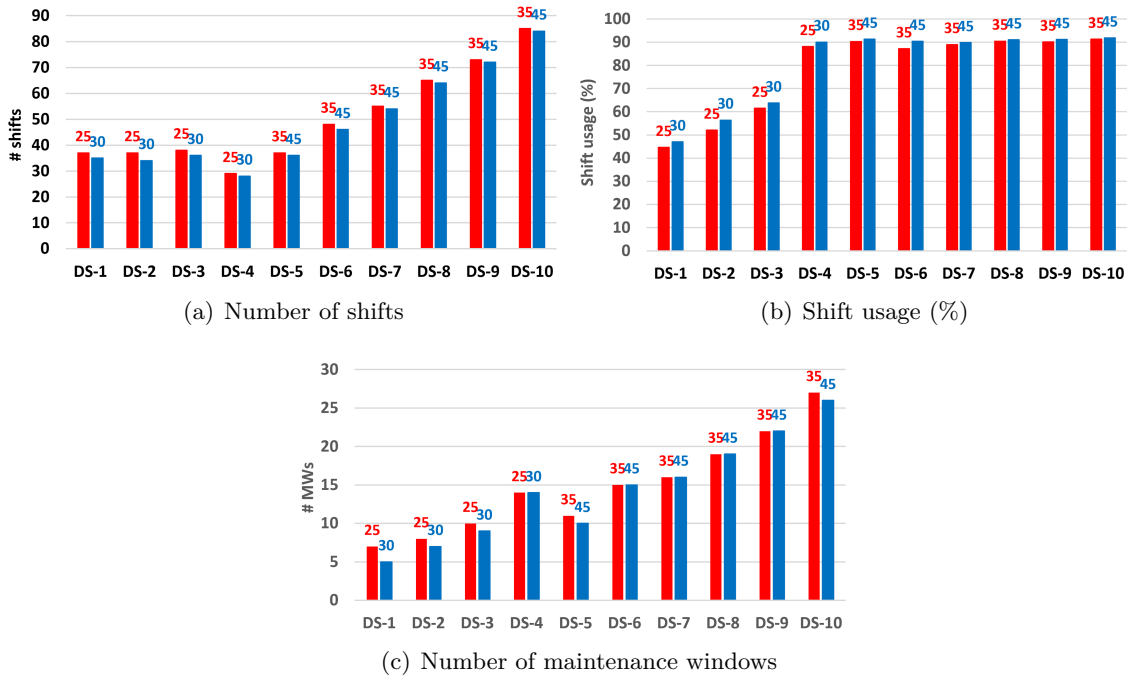


Figure 4.6: Cost analysis of \bar{n}^{CIR} on real data set

Maximum number of migrated circuits per maintenance window (\bar{n}^{CIR}) is a critical parameter as the customers do not like to have frequent disruptions. In Table 4.5, DS-1 to

DS-10 are solved for two different values of \bar{n}^{CIR} . As illustrated in Figure 4.6, we investigate the effect of increasing \bar{n}^{CIR} on number of maintenance windows, number of shifts and the cost of the migration. In Figure 4.6, the value for \bar{n}^{CIR} is written above every column. Figure 4.6(a) shows that an increase in \bar{n}^{CIR} results in the reduction of the number of shifts. This means that having the option to migrate more circuit endpoints per maintenance window gives the model the ability to synchronize more technicians, make more efficient use of shifts and finally reduce the number of shifts. Figure 4.6(b) illustrates how the shift usage increases by increasing \bar{n}^{CIR} . It is interesting to see that an increase in \bar{n}^{CIR} results in less number of shifts and higher shift usage for all instances, which is expected, however, Figure 4.6(c) illustrates that it does not always result in fewer number of maintenance windows. Figure 4.6(c) shows that the number of maintenance windows decreases in 5 instances, while it stays the same in the others. It should be noted that the availability of technicians and engineers are the other limits that can affect the number of maintenance windows. Although increasing \bar{n}^{CIR} can result in fewer shifts, the availability of engineers and technicians can be the issue for decreasing the number of maintenance windows.

As expected, the cost per circuit in the scheduling solution is higher than the planning solution. The reason is that considering exact scheduling constraint that force the both circuit endpoints of the circuits to be migrated in the same time requires longer waiting times. The waiting times can result in longer shifts and in the cases that the maximum duration of shifts is not long enough, Algorithm 1 considers new shifts. Surprisingly, the cost difference between the planning and scheduling solutions is small (less than 2%) in all instances. As seen in Table 4.5, Algorithm 1 can provide a scheduling solution for DS1-DS3 and DS-22 with the same cost as the planning solution's. We should seek the reason in nature of the data sets (Table 4.2). These data sets have either very low or high average number of circuit endpoints distributed over the sites and/or regions. In the scattered data sets with low average number of circuit endpoints, the planning solution has a low shift usage (less than 65%). This means that technicians have some idle time during the planning shifts. Although the algorithm tries to keep technicians working for the whole shift, it is not possible since there are not enough circuits endpoints around. The idle time can be used

Table 4.5: Cost analysis on real data set

Data instance	\bar{n}^{ENG}	\bar{n}^{CIR}	Planning solution				Scheduling solution			Planning vs. Scheduling costs (%)
			# MWs	# shifts	cost per circuit (\$)	shift usage(%)	# MWs	# shifts	cost per # circuit (\$)	
DS-1	2	25	7	37	237.36	44.67	7	37	237.36	0.00
		30	5	35	225.49	47.02	5	35	225.49	0.00
DS-2	2	25	8	37	172.08	52.53	8	37	172.08	0.00
		30	7	34	158.74	56.40	7	34	158.74	0.00
DS-3	2	25	10	38	141.48	61.55	10	38	141.48	0.00
		30	9	36	136.51	63.79	9	36	136.51	0.00
DS-4	2	25	14	29	101.9	88.12	15	31	102.81	0.89
		30	14	28	99.70	90.07	15	30	101.27	1.57
DS-5	2	35	11	37	100.57	90.32	11	37	100.95	0.38
		45	10	36	99.44	91.35	11	38	100.33	0.89
DS-6	2	35	15	48	103.94	87.23	15	48	104.38	0.42
		45	15	46	100.30	90.39	16	48	101.63	1.33
DS-7	2	35	16	55	100.79	88.99	17	57	102.08	1.28
		45	16	54	99.87	89.81	16	54	100.26	0.39
DS-8	2	35	19	65	100.23	90.46	20	67	100.78	0.55
		45	19	64	99.56	91.06	20	66	100.07	0.51
DS-9	2	35	22	73	100.44	90.18	24	77	102.41	1.96
		45	22	72	99.34	91.17	23	76	101.19	1.86
DS-10	2	35	27	85	98.82	91.34	28	87	99.75	0.94
		45	26	84	98.21	91.91	27	86	99.17	0.98
DS-11	2	45	26	94	99.89	89.82	28	98	101.16	1.27
DS-12	2	45	28	100	98.44	91.60	29	102	99.20	0.78
DS-13	3	60	39	149	99.70	89.68	40	151	100.36	0.66
DS-14	3	60	47	193	97.35	90.74	47	193	97.64	0.30
DS-15	3	60	57	237	97.89	90.10	57	237	98.24	0.36
DS-16	3	80	49	281	96.44	91.80	51	287	97.30	0.89
DS-17	3	80	64	346	99.91	88.24	64	346	100.31	0.40
DS-18	3	80	68	378	96.74	91.07	70	382	97.24	0.52
DS-19	4	100	50	399	94.14	95.32	50	399	94.47	0.35
DS-20	4	100	60	462	96.15	93.13	62	470	96.80	0.67
DS-21	4	100	63	495	95.07	94.20	63	495	95.45	0.40
DS-22	4	100	63	499	91.84	98.30	63	499	91.84	0.00

for converting the planning solution to the scheduling solution with no extra costs. On the other hand, for very dense data sets like DS-22, the story is different. The average number of circuit endpoints is high and the planning shifts are almost full (more than 98%). Here, since there are enough number of circuit endpoints to be assigned to the technicians in each site, they don't have to travel between sites and they are synchronized with fewer number of technicians e.g., two technicians are synchronized together the whole shift. In this type of data sets, most of planning shifts are already feasible scheduling shifts i.e., few technicians are synchronized together all the time during the shift with no idle time. For data sets that shift usage is neither too low nor extremely high (85%-95%), the planning shifts are the

combination of both i.e., some with high shift usages and some with idle times. In these cases, since some technicians may be synchronized with several technicians and considering that some of the technicians have no time, changing the order of migrating circuit endpoints for any of the technicians might result in requiring longer shifts or sometimes extra shifts. The difference between costs of the planning and scheduling solutions in Table 4.5 shows that the scheduling solution derived by Algorithm 1 is always less than 2% away from the planning solution and it is less than 1% for 26 out of 32 instances.

4.7 Conclusions

In this work, we developed a branch-and-price algorithm that provides a planning solution for network migration problem by relaxing exact technician scheduling constraints. The algorithm shows promising performance on both artificial and real data instances. We achieved 10% gap between the LP and ILP solutions for all instances within 300 seconds time limit. The planning solution results in a lower bound for network migration problem. We also propose a heuristic to build a scheduling solution based on the obtained planning solution. The results show that the planning solution is an accurate estimate of the scheduling solution such that the planning cost is not more than 2% away from the scheduling cost for the real data sets tested in this paper.

Chapter 5

Wavelength Defragmentation for Seamless Migration

5.1 Introduction

Software Defined Optical Networks (SDONs) can facilitate automation of complex network operations that result in flexibility in the deployment of new services in order to meet changing application requirements. Due to the fact that the lightpaths are set up and torn down more frequently in SDONs, they are likely to become fragmented. This fragmentation leads to a situation in which new requests face a higher blocking probability, even though there is enough capacity to satisfy a demand. In general, wavelength/spectrum defragmentation operations can reduce lightpath blocking probabilities by 3% to 75% [94, 9].

A threefold increase expectation for IP traffic by 2021 [19] is also an evidence of the need for an efficient use of the resources in optical networks. Therefore, it is important to regularly reconfigure the established connections in order to optimize the usage of network resources. *Network reconfiguration* is a term covering different classes of problems, e.g. bandwidth (spectrum) defragmentation in elastic optical networks [104, 105, 106, 55] or capacity recovery in fragmented Multi-Protocol Label Switching (MPLS) networks by rerouting Label Switched Paths (LSP) [54, 56]. In this study, we focus on a subclass of network reconfiguration, namely *wavelength defragmentation problem*.

Wavelength defragmentation, consists of three phases [103, 63]: (i) deciding when to conduct a defragmentation [41]; (ii) designing a new lightpath provisioning with a given optimization objective (e.g., minimum bandwidth requirement) for a given traffic pattern, which ensures the most possible seamless defragmentation; and (iii) migrating from the current fragmented lightpath provisioning to the new optimized one, in the most seamless possible fashion. The focus of our paper is primarily on the third phase, while its difficulty depends on the previous ones. In addition, we study how far we need to consider a sub-optimized lightpath provisioning in order to ensure seamless migration.

In the context of wavelength defragmentation, a network reconfiguration scheme that requires no disruption corresponds to a *Make-Before-Break* (MBB) reachable wavelength provisioning, i.e., a scheme such that we can move from a defragmented wavelength provisioning to an optimized one with the make-before-break process. Within that context, an optimized provisioning is one that reduces significantly the bandwidth requirement, with ideally the smallest possible number of reroutings. Deciding whether there exists an MBB reconfiguration from a current fragmented provisioning to an optimized pre-computed one by rerouting on-going connections one after the other can be done in polynomial time [27]. An MBB reconfiguration can be easily defined using a *Move-To-Vacant* (MTV) algorithm, i.e., sequentially choosing a connection, finding a new path using spare resources and then switching the connection to its new path. The process goes on until no vacant path is found for any of the demands or the new state of the network satisfies the desired constraints e.g., overall usage of resources [69].

In the wavelength defragmentation problem studied in this paper, we aim to find an optimal provisioning for a set of lightpaths such that it can be reached from the current fragmented provisioning with no disruptions (MBB) during the migration.

The paper is organized as follows. Section 5.2 is devoted to literature review on wavelength defragmentation and RWA provisioning algorithms. In Section 5.3, we propose a reconfiguration framework in order to investigate the make-before-break wavelength defragmentation problem. Section 5.4 proposes a nested decomposition optimization algorithm, called WDF_NCG, that computes the minimum bandwidth RWA provisioning, reachable

with make-before-break. Section 5.5 describes the details of WDF_NCG algorithm. Therein, we also propose two other algorithms, one for reducing the size of the dependency graph that records the rerouting ordering of the connections, and one to speed up the restoration of the feasibility conditions at each iteration of the WDF_NCG algorithm. Extensive numerical results are presented in Section 5.6.

5.2 Literature Review

We first review the studies on WDM network reconfiguration, and then the recent work on RWA provisioning. Our proposed algorithms for wavelength defragmentation will borrow some of their ideas.

Note that there are also many references on spectrum or flexible reconfiguration in the context of the Routing and Spectrum Allocation (RSA) problem for flexible optical networks. We omit them as they are not relevant for the WDM network reconfiguration problem studied in this paper, as well as for the models and algorithms we propose.

5.2.1 WDM Network Reconfiguration

Network reconfiguration can be defined as the process of finding out when and how to migrate to a new configuration with a minimum number of disruptions [103]. Network reconfiguration may be required in an optical network due to a change in traffic demand, a failure in the network, a change in the network topology or some maintenance operations [68]. Every network reconfiguration process consists of three phases as described in the introduction.

For the first phase, i.e., when to trigger a reconfiguration, there are several performance metrics, e.g., the average length (number of hops) of the lightpaths [90], the capacity of the alternative routes for a given node pair [102], the granting or denial of a new incoming connection [59].

The second phase consists of building an optimized lightpath provisioning for the current fragmented configuration and its set of requests. We review in Section 5.2.2 the key models

and algorithms of the literature for this step. The third phase defines the order of the reroutings so that the migration is completely seamless or as seamless as possible. This step has been studied with different objective functions, e.g., minimizing the total number of disruptions [53, 20], minimizing the maximum number of concurrent disruptions [25, 27, 24], minimizing disruption time [53] and minimizing reconfiguration costs [8]. A comparison of the first two objective functions can be found in [89, 20]. The second and third phases together build a 2-step approach to reach to a seamless reconfiguration. An alternate option to the second and third phases would be to directly build the best possible provisioning that can be reached without any disruption. While this has been studied numerous times using heuristics, which do not provide any information on how far is their solution from the best possible one, we are aware of only one study [56] with an exact model and algorithm in the context of layer 2 (MPLS) defragmentation.

Seamless WDM network reconfiguration means make-before-break reconfiguration. There are two different ways to achieve this. First way, i.e., the 2-step one as discussed above, is as follows: given the fragmented provisioning and the optimized one, define the best rerouting order so as to find a seamless migration if one exists, or the most seamless possible one. Second way, i.e., a progressive way, is to reroute one connection at a time with, e.g., a Move-To-Vacant (MTV) algorithm, if the goal is to define an MBB reconfiguration. Main drawback of the 2-step way is that very often, no incentive is considered for reducing the number of reroutings when computing the optimized provisioning. Consequently, it often leads to an optimized (e.g., smallest bandwidth requirement or lowest blocking rate) provisioning at the expense of a larger number of connections to reroute. The downside of the progressive way is that it is often carried out with the help of heuristics and therefore no information is available on how far is the resulting provisioning from an optimized (MBB) one.

Seamless wavelength defragmentation has been studied by Palmieri *et al.* [73]. They propose a heuristic based on greedy randomized adaptive search procedure (GRASP). Their GRASP algorithm returns the best MBB provisioning that can be found within the limit

of a given number of iterations. Takita *et al.* [92] propose an Integer Linear Programming (ILP)-based wavelength defragmentation solution for optimizing wavelength resource utilization with minimal optical path disruptions during the migration process. They evaluate their method under a computation time limitation of 10 minutes. The proposed approach can improve the resource efficiency of 20% on sample metro network Japan Photonic Network (JPN) with 48 nodes, 91 bidirectional links and 100 lightpaths. Zhang *et al.* [107] propose different heuristics in order to minimize the disruptions of system resources (transmitters/receivers) while migrating between two given configurations such that any transmitters or receivers (transceivers) in conflict with a lightpath in new provisioning will be torn down. They define a benefit associated with each lightpath e.g., transmission delay or number of conflicts with old lightpaths. They propose different heuristics based on the benefit of a lightpath, for instance, using the same benefit or updating the benefits of remaining unestablished lightpaths. Their proposed heuristics start to build the new provisioning from a lightpath with the largest benefit. They compare the performance of the proposed heuristics based on computation time and number of resource disruptions. The heuristic that considers the number of conflicts as the benefit of a lightpath yields the minimum number of disrupted transceivers among the proposed heuristics.

All reviewed papers in the literature (except [92]) study heuristics and the largest size instances solved so far are on a network (16 nodes and 25 links) with 140 lightpaths [107] and GEANT2 network (33 nodes and 46 bidirectional links) with 1,000 lightpaths [73].

While network reconfiguration with a minimum number of disruptions is certainly of interest in the context of the Internet, most optical network carriers only consider seamless WDM network reconfigurations due to the Service Level Agreements with their customers. Consequently, our objective is to investigate how to build the minimum MBB bandwidth lightpath provisioning for a given defragmentation event, with an exact algorithm.

5.2.2 Routing and Wavelength Assignment

In order to design an efficient wavelength defragmentation algorithm, it is important to review the best models and algorithms for the RWA provisioning. Studies on RWA provisioning differ in their assumptions and objective functions. The most common objective functions are the maximization of the Grade of Service (GoS) or equivalently minimizing the blocking rate (max-RWA) [91, 43, 14, 57] and the minimization of the required bandwidth (min-RWA) [65, 60, 26]. RWA problem has been studied considering two types of traffic: symmetric or undirected connections [44] and asymmetric or directional bandwidth requirements [47]. RWA has also been widely studied for static [57] and dynamic cases [43, 79]. In static cases, the entire set of connections is known in advance. In dynamic cases, a lightpath is set up for each connection request as it arrives. It has been shown in the literature that static RWA is NP-complete [15].

While many heuristics have been proposed to solve the RWA problem [32, 70, 31], significant progress has been made with exact algorithms, allowing to solve exactly the RWA problem for large instances, i.e., networks with up to 90 nodes and 150 wavelengths, and traffic between all node pairs. Jaumard and Daryalal [43] were able solve realistic size instances by proposing an exact algorithm based on a combination of path and link formulations.

5.3 WDM Network Reconfiguration Problem: Our Framework

We present here an overview of the WDM network reconfiguration framework we use in order to investigate the MBB wavelength defragmentation problem. Section 5.3.1 describes the general structure of the 3-step framework. Sections 5.3.2-5.3.3 provide details on each step.

5.3.1 Overview of our WDM Network Reconfiguration Framework

Assume that we are given a WDM optical network, and that the input of our WDM network reconfiguration framework is a demand given by a set of connections. Each connection, if granted, is to be provisioned by a lightpath. A lightpath is a combination of a route and a wavelength. The wavelength should be the same throughout a lightpath, all the way from the source to the destination. This is called continuity constraint. The framework we propose is as follows.

Initialization:

Compute a maximum GoS RWA provisioning, i.e., grant the largest possible number of connections.

Step 1: Dynamic RWA Provisioning

Repeat

Free the resources of each terminating connection.

Grant each incoming connection if there are spare resources to provision it, and else deny it.

Until a wavelength defragmentation is triggered

Step 2: Trigger Defragmentation

When the deterioration of the GoS reaches a given threshold, trigger defragmentation.

Let RWA^{FRAG} be the resulting RWA provisioning.

Step 3: Conduct Defragmentation

Step 3.1. Compute RWA^{OPT} , a minimum bandwidth RWA provisioning for the current state of the network

Step 3.2. Initialize $RWA^{\text{MBB_OPT}}$, with RWA^{OPT}

If $RWA^{\text{MBB_OPT}}$ is MBB reachable from RWA^{FRAG}

Reroute one request at a time using the MBB technique

Return to **Step 1**

Else

Identify some rerouting deadlocks (i.e., conflicting rerouting order such as k needs to be rerouted before k' and vice-versa)

Recompute a minimum bandwidth RWA provisioning with the deadlock avoidance constraints

Let RWA^{OPT} be the new optimized RWA provisioning

Return to **Step 3.2.**

Figure 5.1 illustrates our WDM network reconfiguration framework. For every incoming request, we check whether there are available spare resources to grant it, even if it means routing on a long path. In other words, we search for the shortest lightpath that is available considering the current spare resources. If no spare resource is available, the connection is denied, otherwise it is granted. Better granting proactive algorithms are possible, especially when information is available on the future traffic, but as this is not the focus of this study, we use the simplest rule for granting connections. Hence, for every new connection request we face a dynamic max-RWA problem. For every terminating connection, the resources used by the corresponding lightpath are freed and made available for future traffic.

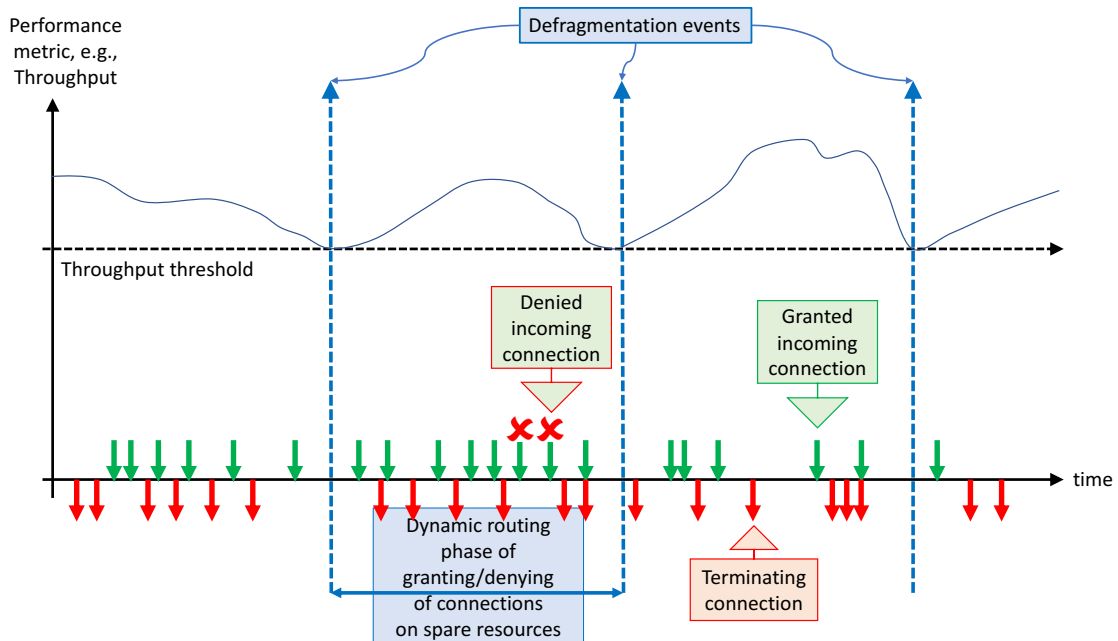


Figure 5.1: WDM Network Reconfiguration Process

5.3.2 Triggering of the Wavelength Defragmentation

While it goes beyond the purpose of this paper to investigate the best way to trigger wavelength defragmentation, we made sure that the way we choose did not facilitate or worsen the wavelength defragmentation, especially under the make-before-break paradigm.

Different performance metrics have been proposed in the literature, see, e.g., [103] for a recent survey on them. We decided to use the Grade of Service (GoS) as the triggering mechanism and, as we will see in the numerical results (see Section 5.6), we use different GoS values, from restrictive one (e.g., a GoS decrease of 5%) to more permissive ones (up to 50%). We also observe that, without wavelength defragmentation, the stabilization of the GoS is reached at various levels, depending on the network topologies, see Section 5.6.

5.3.3 Routing and Wavelength Assignment

Throughout the WDM network reconfiguration framework, we deal with two variants of the RWA problem: Dynamic Max-RWA in the dynamic provisioning phase and Static Min-RWA in the wavelength defragmentation phase. We briefly remind their definitions in what follows.

Static Min-RWA

In static min-RWA, the set of requested connections D is known. The objective function is to grant all connection requests while minimizing the bandwidth requirements over a multigraph $G = (V, L)$ with V (indexed by v) and L (indexed by ℓ) representing the set of nodes and links of G , respectively. $D = (D_{sd})_{(v_s, v_d) \in \mathcal{SD}}$ defines the number of requested unit lightpaths for every node pair, $(v_s, v_d) \in \mathcal{SD} \subseteq V \times V$. Each granted connection is assigned a lightpath (p, λ) where p is a routing path and λ is the selected wavelength among the set of available wavelengths Λ . It should be mentioned that no two lightpaths using the same link can share the same wavelength (under the assumption of a single directional fiber in each direction for connected node pairs).

Dynamic Max-RWA

In dynamic max-RWA problem, the objective function is to grant the largest possible number of connections (GoS) or equivalently to minimize the blocking rate. The set of demands are not known in advance. There is a set of legacy (i.e., on-going) connections D^{LEGACY} routed on a multigraph $G = (V, L)$ and a set of new incoming demands D^{NEW} . We need to assign available lightpaths to D^{NEW} such that no wavelength conflict occurs. Interested reader may refer to [43] and [57] for detailed formulation in both dynamic and static cases of RWA. In the particular case of one new incoming connection to provision at a time, the dynamic max-RWA problem amounts to searching for an available shortest path connecting the two endpoints of the connection.

5.4 A Nested Decomposition Wavelength Defragmentation Algorithm

In this section, we assume that an optimized RWA provisioning, called RWA^{OPT} , is available. We then aim at designing a model (called WDF_MBB) and an algorithm (called WDF_NCG) that will modify RWA^{OPT} as little as possible so that it can be reached with make-before-break from RWA^{FRAG} , the current fragmented RWA provisioning.

One of the key elements of WDF_NCG algorithm is the so-called dependency graph. It is introduced in Section 5.4.1. In Section 5.4.2, we propose a heuristic to reduce the number of the cycles in a dependency graph. The second key element is the computation of an RWA provisioning subject to rerouting deadlock avoidance constraints. Consequently, we review all previous scalable mathematical models for RWA in Section 5.4.3, and select the most relevant one for designing the WDF_MBB model, see Section 5.4.4. Section 5.4.5 describes the WDF_NCG algorithm, which is an iterative algorithm alternating between the solution of the WDF_MBB model and identifying rerouting deadlocks with the dependency graphs, until reaching a minimum bandwidth MBB reachable RWA provisioning.

5.4.1 Dependency Graph and Lightpath Rerouting Order

In order to define the order in which the lightpaths can be rerouted, we build the dependency graph $G_D = (V_D, L_D)$, introduced in [52], between the fragmented (RWA^{FRAG}) and the optimized (RWA^{OPT}) provisionings at the end of each defragmentation interval (see Figure 5.1). A dependency graph is a directed graph that represents the dependence between rerouted requests. Node and link sets of the dependency graph are defined as follows.

$$V_D = \{\pi = (p, \lambda) : \pi \text{ is a lightpath in } RWA^{\text{FRAG}}\}$$

$$L_D = \{(\pi, \pi') : \pi' \text{ needs to be rerouted before } \pi \text{ in order to reach } RWA^{\text{OPT}} \text{ with MBB}\}.$$

In other words, each arc (π, π') defines the order of migration between two lightpaths, when a lightpath π' needs to be rerouted before another lightpath π in order to perform an MBB rerouting.

Figure 5.2 illustrates an example on how to build a dependency graph. As mentioned before, dependency graph is built based on two different provisionings. In Figure 5.2(a) a fragmented provisioning (RWA^{FRAG}) with 29 used links is represented while Figure 5.2(b) shows an optimized provisioning (RWA^{OPT}) for the same set of demands with 21 used links. Figure 5.2(c) is the dependency graph built based on the provisionings presented in Figures 5.2(a) and 5.2(b). Every link in the dependency graph shows one dependency and the right order of rerouting lightpaths. For example, lightpath π_1 in RWA^{OPT} is routed over blue wavelength and has three links ($v_3 \rightarrow v_6$, $v_6 \rightarrow v_5$ and $v_5 \rightarrow v_4$) in common with lightpath π_4 in RWA^{FRAG} . This means that lightpath π_4 needs to be rerouted in order to make room for lightpath π_1 in RWA^{OPT} . In other words, lightpath π_1 cannot be rerouted before lightpath π_4 is rerouted. This dependency (also rerouting order) is represented by a link from vertex π_1 to vertex π_4 in the dependency graph. Table 5.1 shows all the dependencies in this migration. There is one link in the dependency graph for every dependence presented in Table 5.1. It should be mentioned that lightpath π_7 has the same wavelength and the same path in both RWA^{FRAG} and RWA^{OPT} , hence, it has no dependency and does not appear in

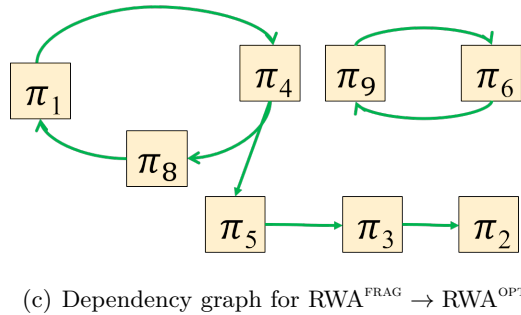
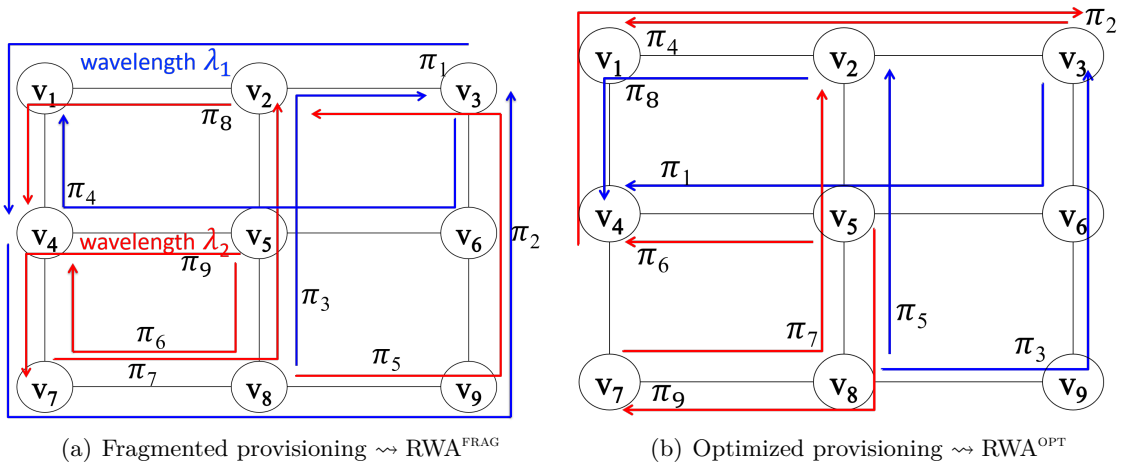


Figure 5.2: An Example of Dependency Graph

the dependency graph.

We use Algorithm 2 to build the dependency graph between RWA^{FRAG} and RWA^{OPT}. For a given lightpath, denote by π_i^{frag} and π_i^{opt} the lightpath in the fragmented and optimized provisionings respectively.

After building the dependency graph, we need to determine the order of rerouting the lightpaths. Since the path and the bandwidth required for the lightpaths with no dependency i.e., no outgoing link, are already available in RWA^{OPT}, they are first rerouted (for example lightpath π_2 in Figure 5.2(c)). After each rerouting, at least one link in the dependency graph is removed and it might result in one or more lightpaths with no dependencies e.g., lightpath π_3 can be rerouted since π_2 is rerouted and it has no other dependencies.

Table 5.1: List of Dependencies

Lightpath	Rerouting dependence(s)	Common links
π_1	π_4	$v_3 \rightarrow v_6, v_6 \rightarrow v_5, v_5 \rightarrow v_4$
π_2	-	-
π_3	π_2	$v_8 \rightarrow v_9, v_9 \rightarrow v_6, v_6 \rightarrow v_3$
π_4	π_5 π_8	$v_3 \rightarrow v_2$ $v_2 \rightarrow v_1$
π_5	π_3	$v_8 \rightarrow v_5, v_5 \rightarrow v_2$
π_6	π_9	$v_5 \rightarrow v_4$
π_8	π_1	$v_2 \rightarrow v_1, v_1 \rightarrow v_4$
π_9	π_6	$v_5 \rightarrow v_8, v_8 \rightarrow v_7$

Algorithm 2 Dependency Graph Builder

- 1: **Input:** Network Topology, RWA^{FRAG} , RWA^{OPT}
 - 2: **Output:** Dependency Graph $G_D = (V_D, L_D)$
 - 3: $V_D \leftarrow \emptyset; L_D \leftarrow \emptyset$
 - 4: **for** every lightpath π_i^{opt} with a different routing from π_i^{frag} **do**
 - 5: $V_D \leftarrow V_D \cup \pi_i$
 - 6: **for** every link $\ell \in L$ **do**
 - 7: **for** every lightpath π_i^{opt} using link ℓ **do**
 - 8: **if** there is a lightpath π_j^{frag} using link ℓ such that $\lambda_{\pi_i^{\text{opt}}} = \lambda_{\pi_j^{\text{frag}}}$ **then**
 - 9: Add arc (π_i, π_j) to L_D
 - 10: **Return** $G_D = (V_D, L_D)$
-

Lightpath π_5 can also be rerouted after lightpath π_3 is rerouted. For the remaining lightpaths, since they are the members of a cycle (a rerouting deadlock), they all have dependencies and rerouting each one requires at least one disruption. There are 2 cycles, one between π_1, π_4 and π_8 and the other one between π_6 and π_9 . Hence, if we want to build an MBB provisioning, we should identify the cycles and prevent them from occurring again. Since maximal link disjoint cycles are easily identifiable once we have computed strongly connected components, we will proceed with the computation of strongly connected components. Several algorithms have been proposed to find the strongly connected components [39] [86], and we use the one proposed by Tarjan [93].

5.4.2 Identification of Avoidable Disruptions: Reducing the Number of Cycles in the Dependency Graph

In this section, we propose an algorithm in order to check whether it is possible to break some of the cycles without adding cuts and compromising the optimal solution by finding new lightpaths for one or some of the connections. Figure 5.3 represents an example of removing a cycle by finding a new lightpath for π_1 . As seen in Figure 5.3(c), there is a cycle between π_1 and π_2 while migrating from the fragmented provisioning in Figure 5.3(a) to the optimized one in Figure 5.3(b). If we can find a new lightpath for one of these dependencies without creating new cycles, then a seamless migration will be possible. Figure 5.3(d) shows that a new lightpath can be assigned to lightpath π_1 that not only does not create a new cycle but also breaks the cycle between π_1 and π_2 . It should be mentioned that it is also possible to find a new lightpath for π_2 and get the same results e.g., changing the wavelength of π_2 from blue to red on the same path in Figure 5.3(d) returns the same result that is an MBB provisioning.

In Algorithm 3, after building the dependency graph G_D and finding the set C of strongly connected components (SCCs), for each lightpath involved in an SCC, we try to find a new lightpath, of the same length, that allows the break of a cycle in the SCC and eventually make it acyclic. If such a new lightpath is found for a given connection request, we use it and update G_D accordingly. We repeat this process until G_D becomes acyclic or we can no longer find any new lightpath helping to break the cycles. If there are no more cycles, it means that an MBB provisioning has been found. But, finding no lightpaths to be rerouted while there are still some cycles shows that Algorithm 3 cannot break any other cycles and a seamless migration is not yet possible.

It should be mentioned that the nodes in G_D are prioritized based on their in-degree (the number of arcs ending in each node). The reason is that having higher in-degree for a given lightpath π shows that more lightpaths are dependent on π and it can contribute to more cycles.

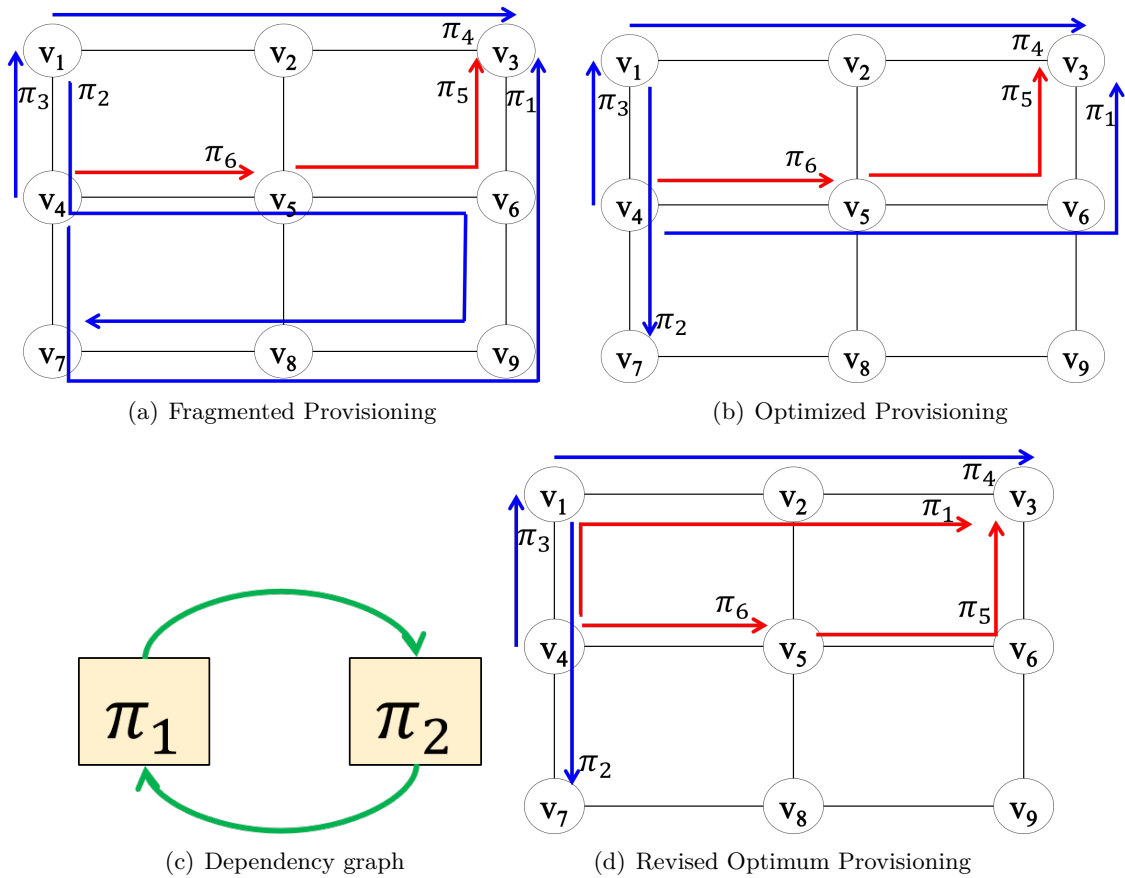


Figure 5.3: Preprocessing Operation

5.4.3 Comparison of the RWA Decomposition Models

As mentioned before, we need to use static min-RWA in order to efficiently take advantage of the network capacities. Table 5.2 explains four different decompositions for static min-RWA and the required modifications in order to consider the network reconfiguration problem.

In lightpath decomposition, each configuration (variable) z_γ represents a lightpath between a node pair. The biggest advantage is that each pricing problem can be solved using a polynomial algorithm e.g., Dijkstra's algorithm. The number of pricing problems are dependent on the network size and we will face a huge number of pricing problems as the network size grows. Cycle elimination constraints are in link level and simple in compare to other decompositions (See Table 5.2). They need to be added to the master problem as

Algorithm 3 Preprocessing Algorithm

Input: Dependency graph G_D

Input: $G_\lambda = (V, L_\lambda) \subseteq G = (V, L)$ where L_λ is the set of links which are not used in any lightpath using λ , for $\lambda \in \Lambda$.

Output: New wavelength assignment for some lightpaths involved in cycles

- 1: Computes the strongly connected components (SCCs) of G_D
 - 2: Build the list Π of lightpaths, i.e, the set of nodes in G_D , involved in the SCCs
 - 3: Sort Π by non-increasing in-degree in G_D
 - 4: **for** every node (lightpath) $\pi = (p, \lambda) \in \Pi$ such that $p : v_s \rightsquigarrow v_d$ **do**
 - 5: **for** every $\lambda' \in \Lambda$ **do**
 - 6: $\hat{p} \leftarrow$ shortest path from v_s to v_d in $G_{\lambda'}$
 - 7: **if** $\hat{p} \neq \emptyset$ and $|\hat{p}| \leq |p|$ **then**
 - 8: Check if changing π to (\hat{p}, λ') creates
 - 9: new cycles in G_D
 - 10: **if** no new cycle is generated **then**
 - 11: Set $\pi = (\hat{p}, \lambda')$
 - 12: Update G_λ and $G_{\lambda'}$
 - 13: Go to step 1
 - 13: **return** Set of lightpaths
-

each pricing problem corresponds to just one lightpath. The constraints directly limit the usage of undesirable lightpaths.

In wavelength decomposition, each variable z_γ is a set of link disjoint paths on the same generic wavelength. This is the reason why this decomposition is the only one that does not suffer from the symmetry. It has only one pricing problem that is big and complicated. Cycle elimination constraints are in the configuration level and limit the configurations that build a cycle. The same cycle might be regenerated in the forthcoming configurations.

In the indexed wavelength decomposition, the cycle elimination constraints might also be added to the pricing problem if all lightpaths building the cycle are on the same wavelength. Since there is one pricing problem per wavelength in this decomposition, the usage of the undesirable lightpaths can be restricted inside the corresponding pricing problem. The existing approaches use a link formulation and speed it up by adding a path formulation. We propose a new decomposition scheme for indexed wavelength decomposition based on nested column generation method. In this approach, a set of precomputed paths are used in the path formulation and the longer paths are generated only when they are needed and with respect to node pairs.

Table 5.2: Comparison of 4 decompositions

	Lightpath	Wavelength Plan	Indexed Wavelength Plan	
	Wavelength indexing	No wavelength indexing \rightsquigarrow requires a wavelength assignment	Wavelength indexing	
(Restricted) Master Problem				
Variable definition	A lightpath for a node pair	A set of link disjoint paths on the same generic wavelength	A set of link disjoint paths on the same specific wavelength	A set of link disjoint paths on the same specific wavelength
# of basis variables	$\sum_{(v_s, v_d) \in \mathcal{SD}} D_{sd}$	$ \Lambda $	$ \Lambda $	$ \Lambda $
# of constraints	$ \mathcal{SD} + \Lambda L $	$1 + \mathcal{SD} $	$ \Lambda + \mathcal{SD} $	$ \Lambda + \mathcal{SD} $
Symmetry	Wavelength permutation leads to an equivalent (optimal) solution	No symmetrical solution	Wavelength permutation leads to an equivalent (optimal) solution	Wavelength permutation leads to an equivalent (optimal) solution
Cycle elimination constraints	Master problem in link level (constraints can define which combination of links and wavelength may not be used at the same time)	<p>Master problem in configuration level</p> <ul style="list-style-type: none"> Constraints are imposed on already generated configurations and restricts the usage of a set of undesirable columns together. Later configurations are generated under no constraint in order to avoid creating cycles 	<ul style="list-style-type: none"> Master problem: If connections building the cycle are routed on different wavelengths (link level) Pricing problem: If all connections building a cycle are routed on the same wavelength 	<ul style="list-style-type: none"> Master problem: If connections building the cycle are routed on different wavelengths (link level) Pricing problem: If all connections building a cycle are routed on the same wavelength
Mathematical expression	<p>Configuration indexed by $\pi \equiv$ lightpath</p> $\sum_{\pi \in c} z_\pi \leq c - 1, c \in C$ <p>Pros 1. Since a configuration is equivalent to a lightpath, current lightpaths building the cycle will not be generated again.</p> <p>Pros 2. Each constraint directly limits the usage of undesirable lightpaths, not a set of lightpaths of which the undesirable one is a member.</p>	<p>Configuration indexed by $\gamma \equiv$ wavelength plan for a generic wavelength</p> $\sum_{\gamma \in \Gamma: \pi \in \gamma \cap c} z_\gamma \leq c - 1, c \in C$ <p>Cons 1. Prevent from using a given cycle for given configurations, but does prevent from regenerating the same cycle with forthcoming configurations.</p> <p>Cons 2. Eliminate a cycle indirectly throughout forbidding the subset of configurations containing the lightpaths involved in the cycle</p> <p>Cons 3. If ILP solution derived by solving exactly the last RMP, higher probability of infeasibility</p>	<p>Configuration indexed by $\gamma \equiv$ wavelength plan for a given λ</p> $\sum_{\lambda \in \Lambda} \sum_{\gamma \in \Gamma_\lambda} \left(\sum_{\pi \in c} a_\pi^\gamma \right) z_\gamma \leq c - 1 \quad c \in C$ <p>Cons 1. Solving Link formulation is long for large networks.</p> <p>Pros 1. Generate longer paths/lightpaths only as needed, and selectively wrt sd</p>	
Pricing Problem(s)				
Pricing Problems	Polynomial: Weighted shortest path	Finding a set of link disjoint paths: NP-complete [62]		
	link formulation	link formulation	link formulation	path formulation & nested column generation
Number of pricing problems	$ \Lambda \mathcal{SD} $	1	$ \Lambda $	$ \Lambda $
Size of pricing problems	<p>Small and simple. 1 type of pricing problem is required:</p> <ul style="list-style-type: none"> A polynomial-time algorithm (Dijkstra's algorithm) 	<p>Big and complicated. 1 type of pricing problems is required:</p> <ul style="list-style-type: none"> Link formulation <p>Adding the second type of pricing problem makes algorithm faster</p> <ul style="list-style-type: none"> Path formulation: Using a set of precomputed paths in order to generate a configuration. 	<p>Big and complicated. 1 type of pricing problems is required:</p> <ul style="list-style-type: none"> Link formulation <p>Adding the second type of pricing problem makes algorithm faster</p> <ul style="list-style-type: none"> Path formulation: Using a set of precomputed paths in order to generate a configuration. 	<p>1 path formulation and a nested column generation are required:</p> <ul style="list-style-type: none"> Path formulation: Uses a set of precomputed paths. Nested column generation: Generates generic paths if needed.

5.4.4 WDF_MBB Model

As seen in Section 5.2.2, RWA problem has been widely studied and there are several exact solution schemes in the literature including compact and decomposition formulations. Although the compact formulations are not scalable, real size instances can still be solved exactly using the decomposition methods e.g., column generation technique. According to the recent RWA studies [43, 66], the most efficient decomposition is to generate wavelength plans (i.e., set of pairwise link disjoint lightpaths using the same wavelength) using a path formulation, followed by a link formulation in order to guarantee the optimality of the linear relaxation of the model.

In order to avoid the use of a link formulation, which is computationally costly to solve, we propose to use a nested decomposition scheme for designing an efficient mathematical model for the wavelength defragmentation problem. In this section, we first establish the so-called master problem of the decomposition scheme, called WDF_MBB model, and will next discuss its detailed solution in Section 5.5.

The WDF_MBB model relies on the concept of wavelength configurations, where a configuration γ corresponds to a wavelength plan, for a given wavelength λ . While in RWA decomposition models (see Table 5.2), configuration can be defined for a generic wavelength and therefore do not have any symmetry issues (the wavelength assignment is done a posterior), this is unfortunately not possible here in order to be able to express the rerouting deadlock avoidance constraints.

Model WDF_MBB requires one unique set of decision variables z_γ for $\gamma \in \Gamma$. z_γ is a binary variable deciding on the selection of configuration γ in the optimal RWA provisioning output.

Each configuration $\gamma \in \Gamma$ is formally defined by the following parameters:

- B^γ Bandwidth requirement of configuration γ , as expressed by the number of links used in γ for routing some connections.
- a_{sd}^γ Number of lightpaths serving node pair (v_s, v_d) in configuration γ .
- a_π^γ = 1 if lightpath π is used in configuration γ , 0 otherwise.

We now express the WDF_MBB that computes the minimum bandwidth RWA provisioning subject to deadlock avoidance constraints as identified in the dependency graph. Since the number of cycles in a directed graph can be exponential, we do not introduce all possible cycles. By using Tarjan’s algorithm, we find strongly connected components. While each strongly connected component can contain several cycles, we add only one cycle per strongly connected component in order to control the number of constraints, i.e., the cycle that includes all nodes of the strongly connected component.

$$\text{Minimize: } \sum_{\gamma \in \Gamma} B^\gamma z_\gamma \quad (5.1)$$

$$\text{Subject to: } \sum_{\gamma \in \Gamma_\lambda} z_\gamma \leq 1, \quad \lambda \in \Lambda \quad (5.2)$$

$$\sum_{\gamma \in \Gamma} a_{sd}^\gamma z_\gamma \geq D_{sd}, \quad (v_s, v_d) \in \mathcal{SD} \quad (5.3)$$

$$\sum_{\gamma \in \Gamma} \left(\sum_{\pi \in c} a_\pi^\gamma \right) z_\gamma \leq |c| - 1, \quad c \in C \quad (5.4)$$

$$z_\gamma \in \{0, 1\}, \quad \gamma \in \Gamma. \quad (5.5)$$

Constraint set (5.2) ensures that we select at most one configuration for each wavelength λ . Constraint set (5.3) enforces the demand constraints, with the left-hand side term computing the number of demand units provided by each configuration, and then summing over all configurations. Constraint set (5.4) guarantees that the provisioning will not contain any of the cycles $c \in C$. Constraint set (5.5) defines the domains of the variables.

5.4.5 WDF_NCG Algorithm

In order to find an MBB provisioning, we propose an iterative process presented in Algorithm 4. This process starts with building a dependency graph G_D for migrating from a fragmented provisioning (RWA^{FRAG}) to an optimized provisioning (RWA^{OPT}). Tarjan’s algorithm is used in order to find the set of cycles C (strongly connected components) in G_D . Algorithm 3 helps us break some of the cycles. If C is empty, it means that a seamless migration has been found. Otherwise, one constraint corresponding to each cycle $c \in C$

is added to WDF_MBB . The solution of WDF_MBB is the updated RWA^{OPT} that does not contain any of the cycles found in previous iterations due to deadlock avoidance constraints. However, it may contain some new cycles. Hence, the dependency graph is generated, set of cycles are found and Algorithm 3 tries to reduce the number of cycles in C . This process goes on until the set of cycles C is empty.

Algorithm 4 WDF_NCG Algorithm

Input: RWA^{FRAG} , RWA^{OPT}

Output: make-before-break RWA^{OPT}

- 1: Build dependency graph G_D (Algorithm 2).
 - 2: Find the set of cycles C in G_D (Tarjan's Algorithm).
 - 3: Reduce the number of cycles C (Algorithm 3).
 - 4: **while** C is not empty **do**
 - 5: Update RWA^{OPT} by solving WDF_MBB
 - 6: Build dependency graph G_D (Algorithm 2).
 - 7: Find the set of cycles C in G_D (Tarjan's Algorithm).
 - 8: Reduce the number of cycles C (Algorithm 3).
 - 9: **return** RWA^{OPT}
-

5.5 Solution of WDF_MBB Model: A Nested Decomposition Wavelength Defragmentation Model

We first discuss the details of the solution process for the WDF_MBB model with a nested decomposition algorithm in Section 5.5.1. We next discuss in Section 5.5.2 an algorithm to speed up the feasibility in WDF_MBB model after the addition of new rerouting deadlock avoidance constraints.

5.5.1 Nested Column Generation

The WDF_MBB model proposed in Section 5.4.4 has an exponential number of variables, and therefore is not scalable if solved using classical ILP tools. Indeed, we need to use column generation techniques in order to manage a solution process that only requires an *implicit* enumeration of the wavelength configurations (interested readers may refer to

Chvátal [16]). Column generation method allows the exact solution of the linear programming relaxation of model (5.1)-(5.5), where variables $z_\gamma \in Z^+$ are replaced by $z_\gamma \geq 0$, for $\gamma \in \Gamma$. It consists of solving alternatively a restricted master problem (the WDF_MBB model in Section 5.4.4 with a limited number of columns/variables) and the pricing problem (generation of a new wavelength configuration) until the optimality condition is satisfied (i.e., no wavelength configuration with a negative reduced cost is found). In other words, when a new wavelength configuration is generated, it is added to the current restricted master problem only if its addition implies an improvement of the optimal value of the current restricted master problem. This condition, indeed an optimality condition, can be easily checked with the sign of the reduced cost of variables z_γ , denoted by $\overline{\text{COST}}$, see (5.6) for its expression (the reader who is not familiar with linear programming concepts is referred to [16]).

Once the optimal solution of the LP relaxation (z_{LP}^*) has been reached, we solve exactly the last restricted master problem with integer variables using a branch-and-bound method, leading then to an ε -optimal ILP solution (\tilde{z}_{ILP}), where

$$\varepsilon = \frac{z_{\text{LP}}^* - \tilde{z}_{\text{ILP}}}{z_{\text{LP}}^*}.$$

Branch-and-price methods can be used in order to find optimal solutions, if the accuracy (ε) is not satisfactory, see, e.g., [4, 45].

In order to solve WDF_MBB, we apply a nested column generation algorithm in which each pricing problem is solved by column generation itself. In other words, the master problem comprises all generated wavelength configurations. Each wavelength configuration is a set of feasible link disjoint paths and pricing problems are responsible for generating them. The process of generating and selecting a set of link disjoint paths is solved by column generation method in each pricing problem. Figure 5.4 represents a nested column generation scheme.

Each pricing problem includes a nested master problem and a set of nested pricing problems. Each nested pricing problem generates a path for a given node pair $(v_s, v_d) \in$

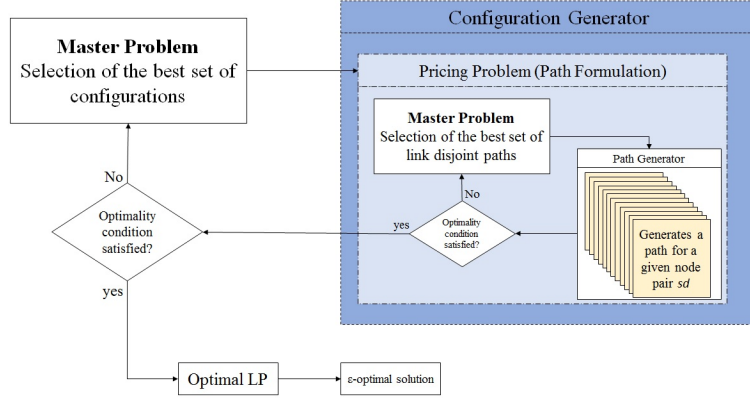


Figure 5.4: Nested Column Generation

SD . The objective in the nested master problem represents the cost of the wavelength configuration in the current pricing problem. After introducing variable β_p^{sd} used in this section, we will present the nested master and nested pricing problems.

$\beta_p^{sd} = 1$ if path p is used in the wavelength configuration under construction, 0 otherwise.

Note that γ is omitted in the sequel in order to alleviate the notations (e.g., $B^\gamma \rightsquigarrow B$).

$$\min \quad \overline{\text{COST}}^{\text{CONFIG}} = B - u_\lambda^{(5.2)} - \sum_{(v_s, v_d) \in SD} \sum_{p \in P_{sd}} \beta_p^{sd} u_{sd}^{(5.3)} \quad (5.6)$$

subject to:

$$\sum_{(v_s, v_d) \in SD} \sum_{p \in P_{sd}} \delta_\ell^p \beta_p^{sd} \leq 1 \quad \ell \in L \quad (5.7)$$

$$\sum_{p \in P_{sd}} \beta_p^{sd} \leq D_{sd} \quad (v_s, v_d) \in SD \quad (5.8)$$

$$\sum_{(v_s, v_d) \in SD} \sum_{p \in P_{sd}} \sum_{\ell \in L} \delta_\ell^p \beta_p^{sd} \leq B \quad (5.9)$$

$$\beta_p^{sd} \in \{0, 1\} \quad (v_s, v_d) \in SD, p \in P_{sd}. \quad (5.10)$$

where δ_ℓ^p is equal to 1 if link ℓ belongs to path p .

Constraint set (5.7) assures that selected paths are link disjoint. Thanks to constraint

set (5.8), no more than D_{sd} is granted for every node pair $(v_s, v_d) \in \mathcal{SD}$. Constraint set (5.9) determines the number of used links in the current configuration. Observe that for a path $p \in P_{sd}$, $\sum_{\ell \in L} \delta_\ell^p$ is the length of path p .

In order to generate a path between a given node pair $(v_s, v_d) \in \mathcal{SD}$, we find the path with minimum cost in the nested pricing problem:

$$\min \quad \overline{\text{COST}}^{\text{PATH}} = -u_{sd}^{(5.3)} - \sum_{\ell \in L} \delta_\ell u_\ell^{(5.7)} - u_{sd}^{(5.8)} - u^{(5.9)} \sum_{\ell \in L} \delta_\ell \quad (5.11)$$

Since $u_{sd}^{(5.3)}$ and $u_{sd}^{(5.8)}$ are constant values for a given node pair, equation (5.11) can be rewritten as

$$\min \quad \overline{\text{COST}}^{\text{PATH}} = - \sum_{\ell \in L} \delta_\ell u_\ell^{(5.7)} - u^{(5.9)} \sum_{\ell \in L} \delta_\ell \quad (5.12)$$

As $u_\ell^{(5.7)}$ and $u^{(5.9)}$ come from inequality constraints that are “ \leq ”, and that we have a minimization problem, they are both non-positive. This means that we can assign a non-negative weight $-u_\ell^{(5.7)} - u^{(5.9)}$ to each link and use a polynomial-time algorithm, e.g., Dijkstra’s algorithm to find a weighted shortest path for a given node pair.

5.5.2 Re-Establishment of a Feasible RWA Provisioning

We propose Algorithm 5 in order to overcome the infeasibility while obtaining an ILP solution. The algorithm is based on the configurations generated in the nested column generation algorithm and uses the configurations with greatest LP values for building a feasible ILP solution. Algorithm 5 starts with a set of feasible (optimum) LP configurations (Γ_{LP}). The configurations are sorted based on their LP values and the configuration with maximum value for every wavelength λ is selected and added to the set of ILP configurations (Γ_{ILP}). As the demand between some node pairs might be overfilled in the initial Γ_{ILP} , the longest lightpaths assigned to overfilled demands are removed and the corresponding links are released to be used later. Then, all cycle constraints are checked and if some of them are violated, longest lightpaths are removed in order to satisfy the constraints. In next

step, the set D_{uf} of demands that are not fulfilled yet is found. The algorithm first tries to answer the unfulfilled demands by using the free links such that all cycle constraints are satisfied. If it is not possible, it makes some changes and frees up capacity for a path by removing the longest paths from busiest configurations. It should be mentioned that cycle constraints are always checked while assigning a new lightpath to a demand. The process goes on until all demands are fulfilled.

Algorithm 5 Building a feasible ILP solution

Input: A feasible LP solution: A set of configurations Γ_{LP}

Output: A feasible ILP solution: A set of configurations Γ_{ILP}

- 1: $\Gamma_{ILP} = \emptyset$.
 - 2: Sort configurations in LP based on their LP value.
 - 3: **for** $\lambda \in \Lambda$ **do**
 - 4: Choose a configuration γ with max LP value.
 - 5: $\Gamma_{ILP} = \Gamma_{ILP} \cup \{\gamma\}$
 - 6: Update Γ_{ILP} by removing lightpaths related to overfilled demands (longest lightpaths).
 - 7: Update Γ_{ILP} by removing lightpaths violating the cycle constraints (longest lightpaths).
 - 8: Build the set of unfulfilled demands (D_{uf}).
 - 9: **while** D_{uf} is not empty **do**
 - 10: Choose a demand $d \in D_{uf}$
 - 11: **if** It is possible to find a lightpath π satisfying cycle constraints **then**
 - 12: Assign π to d
 - 13: $D_{uf} = D_{uf} - \{d\}$
 - 14: **else**
 - 15: Find the busiest configuration γ (max # of lightpaths)
 - 16: Free up the shortest path for d in γ by removing a set of lightpaths related to a set of demands D_r
 - 17: $D_{uf} = D_{uf} - d$
 - 18: $D_{uf} = D_{uf} \cup D_r$
 - 19: **return** Γ_{ILP}
-

5.6 Computational Results

5.6.1 Traffic and Network Data Sets

We use four different networks, whose key characteristics (number of nodes and links, average node degrees) are described in Table 5.3.

Table 5.3: Main characteristics of the networks

Networks	$ V $	$ L $	Avg. deg.	# wavelengths
USA [6]	24	88	3.7	75
GERMANY [72]	50	176	3.5	130
NTT [101]	55	144	2.6	42
CONUS [88]	60	158	2.6	30 & 50

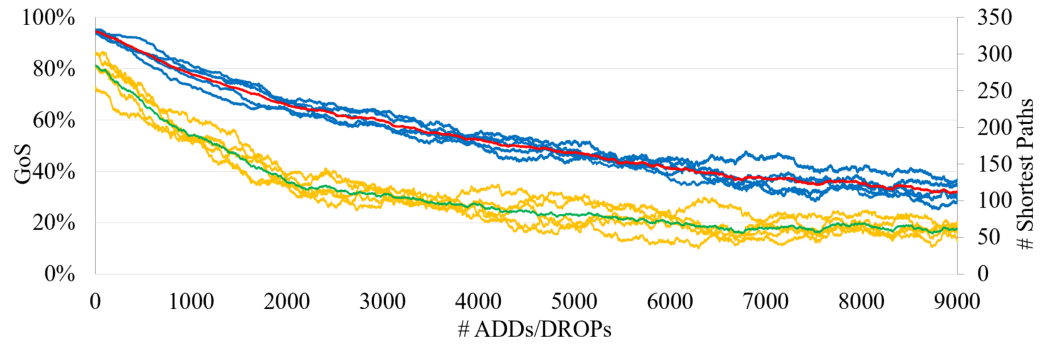
Table 5.4: Statistics for the Characteristics of the Strongly Connected Components in the Dependency Graphs

Networks	GoS Reduction	# of nodes in SCC/AVG degree (# Edges/# Nodes)										# SCCs									
		SCC #1	SCC #2	SCC #3	SCC #4	SCC #5	SCC #6	SCC #7	SCC #8	SCC #9	SCC #10										
NTT	10%	282	2.20	3	2	2	1	2	1	1	0										589
	15%	251	2.23	3	1.33	1	0														581
	20%	318	2.28	3	3.33	2	3	1	0												445
	25%	170	2.46	6	1.83	3	2	3	1.67	3	1.33	2	6.50	2	3	2	1.50	1	0		542
USA	5%	529	2.13	1	0																412
	10%(I)	506	2.11	2	3	1	0														387
	10%(II)	427	2.01	1	0																468
	15%	428	2.11	3	6	2	3	2	1	1	0										413
GERMANY	5%	1,062	2.43	1	0																775
	10%	544	2.16	3	2	3	2	2	2.5	1	0										1,190
	15%	175	1.65	5	1.60	3	2	2	2	1.50	2	1.5	2	1.50	2	1	1	0			1,453
	20%	20	1.50	10	1.40	5	1.60	3	2	2	1.50	2	1	1	0						1,413
	25%	16	2	1	0																1,446
CONUS (30)	15%	136	2.91	7	2.57	3	2	2	1.50	1	0										240
	20%	93	2.80	7	2.50	2	2	1	0												261
	30%	142	2.74	4	1.75	3	1	1	0												175
	40%	122	2.98	2	2.00	2	1	1	0												148
	50%	44	1.97	2	1.50	1	0														188
CONUS (50)	25%	249	2.73	9	2.25	7	2.33	4	2	1	0										469
	30%	153	2.31	9	2.44	4	2	2	1	1	0										504
	35%	184	2.64	2	4	2	3	2	1	1	0										444

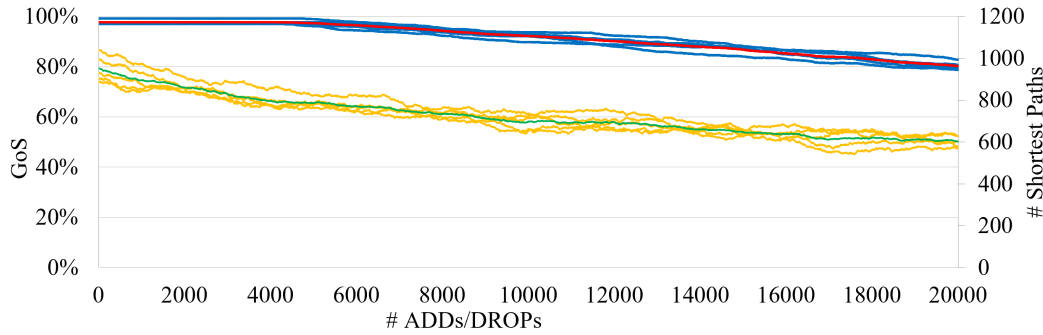
5.6.2 Generation of Fragmented RWA Provisionings

In order to generate an RWA provisioning, we use the framework described in Section 5.3. First initial provisioning corresponds to the RWA provisioning obtained in Jaumard and Daryalal [43], with the objective of maximizing the GoS. We then use a dynamic RWA process with ADD and DROP requests, using a random generator such that the probability of ADD and DROP is the same, i.e., 0.5. Source and destination of the ADD and DROP requests are selected at random.

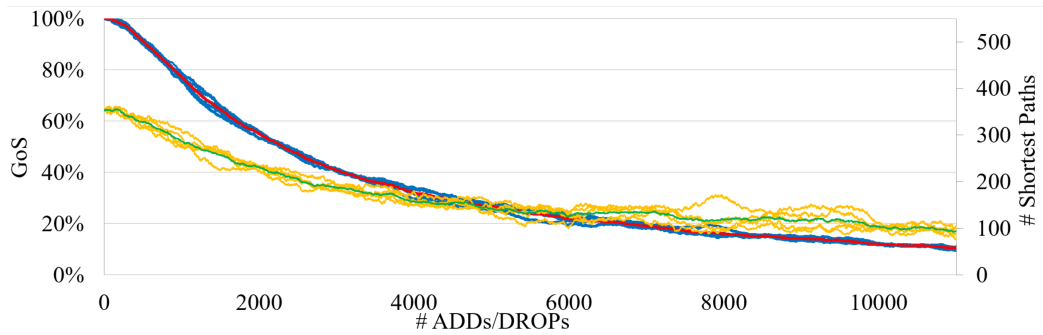
We report the GoS behavior for each network in Figure 5.5, where the GoS (vertical axis) is expressed with the number of granted requests. We plot two sets of curves, the top ones (blue curves) are each associated with one random dynamic traffic and the red curve is the average of the five individual dynamic RWA curves. The second set of curves (bottom



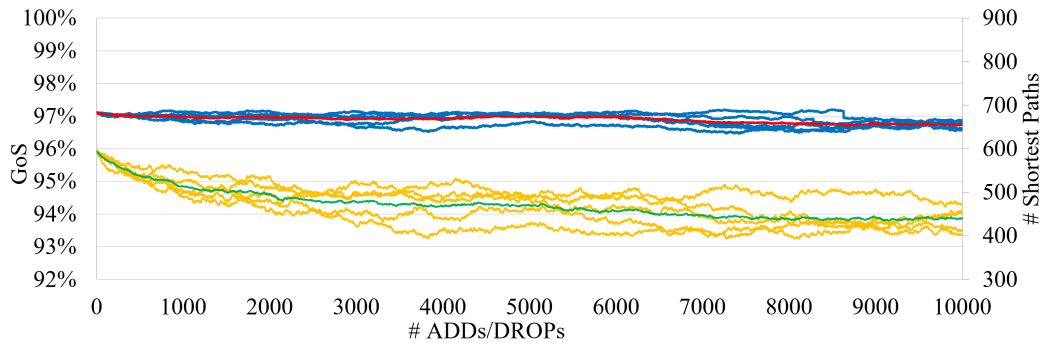
(a) NTT



(b) Germany



(c) CONUS



(d) USA

— GoS — Average GoS — # Granted Requests on SP — Average # of Granted Requets on SP

Figure 5.5: Fragmented Networks (GoS)

ones) corresponds to the number of granted requests on a shortest path (be aware that for a given node pair, there may exist several shortest paths). Again, we have 5 individual curves (orange ones) and one additional curve (green curve) that represents their average.

We observe that it takes a variable number of ADD and DROP requests before reaching a steady state, depending on the networks. In addition, the steady state is reached for different values of GoS. USA (Figure 5.5(d)) and GERMANY (Figure 5.5(b)) networks have a larger GoS value than NTT (Figure 5.5(a)) and CONUS (Figure 5.5(c)) in their steady states. An explanatory factor is the average degree of the networks, see Table 5.3. Indeed, USA and GERMANY have a higher average degree, and consequently are more connected. This leads to more choices for establishing a lightpath for incoming requests, with the length of routes increasing more slowly. Behavior of the number of requests routed on a shortest path follows the behavior of the GoS decrease, as expected. As the GoS decreases, the network becomes more fragmented and, as a result, fewer shortest paths are available to grant and route incoming requests.

Table 5.5: Statistics for the SCCs after preprocessing algorithm

Networks	GoS Reduction	# of nodes in SCC/AVG degree (# Links/# Nodes)								# SCCs				
		SCC #1	SCC #2	SCC #3	SCC #4	SCC #5	SCC #6	SCC #6						
NTT	10%	2	2.50	1	0					874				
	15%	20	1.85	1	0					814				
	20%	16	2.25	9	2.00	3	1.67	2	1.00	2	3.50	1	0	738
	25%	3	1.67	3	1.67	2	7.50	2	1.50	2	1.00	1	0	722
USA	5%	4	2.00	4	1.75	2	1.50	2	1.50	1	0			932
	10%(I)	24	2.00	1	0									870
	10%(II)	7	1.14	1	0									888
	15%	7	1.57	5	1.20	1	0							834
GERMANY	5%	1	0											1,836
	10%	9	1.45	3	2.00	1	0							1,728
	15%	16	1.44	2	1.50	2	1.50	1	0					1,621
	20%	5	1.60	2	1.50	2	1	1	0					1,446
	25%	1	0											1,453
CONUS (30)	15%	4	2.50	2	1.50	2	1.00	1	0					379
	20%	12	1.91	1	0									351
	30%	6	2.16	2	1.00	1	0							307
	40%	7	3.00	6	2.16	2	4.00	2	2.50	2	1.00	1	0	237
	50%	1	0											232
CONUS (50)	25%	5	3.44	4	1.5	2	2	1	0					714
	30%	2	1.50	1	0									667
	35%	2	3	2	1.50	1	0							629

5.6.3 Reduction of the Number of Rerouting Deadlocks

Distribution and Sizes of the Strongly Connected Components

As mentioned in Section 5.2.1, network reconfiguration is triggered by the GoS. Table 5.4 shows the different GoS values we considered. For every GoS value, we computed the strongly connected components of the dependency graph. If they are all of size one, we can conclude that the optimal RWA provisioning is MBB reachable. However, this was the case for none of the data instances as shown in Table 5.4: each data instance is such that its associated dependency graph has at least one strongly connected component with a size larger than 1.

We observe that most data instances are such that the dependency graph contains a single strongly connected component with more than one node. In addition, this component is usually pretty large, as could have been anticipated by the theoretical results about the distribution and the sizes of strongly connected components in random graphs, see, e.g., [21].

Reduction of the Number of Rerouting Deadlocks

We report in Table 5.4 the number and sizes of the strongly connected components after applying Algorithm 3. We observe that the number of strongly connected components has significantly increased as a consequence of the drastic reduction of the size of the largest strongly connected components, still on average larger than one. While seamless migration is still not yet possible for most of the cases except for GERMANY (5%) and CONUS (50%), the number of rerouting deadlocks, as expressed by the number of cycles, and hence the size and the density of the strongly connected components of size larger than 1, has significantly decreased.

Reduction of the size of the biggest strongly connected component for each data instance is represented in Figure 5.6.

Figure 5.7 illustrates the efficiency of using the ordering criterion of Algorithm 3 with respect to the in-degree of the nodes of the dependency graph. Figure 5.7(a) represents

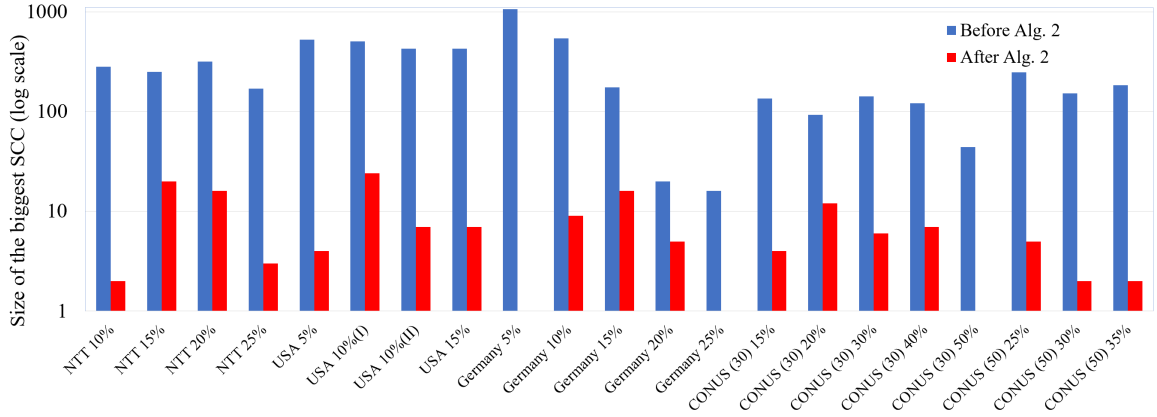


Figure 5.6: Performance of Algorithm 3: Reduction of the size of the biggest strongly connected component.

the strongly connected component of the USA (5%) data set, after the first construction of the dependency graph. Therein, we can observe many rerouting deadlocks. Figure 5.7(b) depicts the largest strongly connected component after applying Algorithm 3, using an arbitrary order of the nodes: the size of the largest strongly connected component has significantly decreased, and we can observe a weak density with several interconnected cycles. Lastly, Figure 5.7(c) shows again the largest strongly connected component, this time after applying Algorithm 3 with its current ordering rule for node processing. We can note a very significant reduction of the strongly connected component, now limited to a single cycle, i.e., a single rerouting deadlock.

5.6.4 Seamless or Almost Seamless Wavelength Defragmentation

We now evaluate the compromise to be made on the minimization of the bandwidth requirement in order to get a seamless wavelength reconfiguration. As described in Section 5.4.5, the WDF_NCG algorithm searches for alternate wavelength provisioning when a rerouting deadlock is identified in the dependency graph, and it may result in increasing the bandwidth requirements.

We report the results in Table 5.6. No data instance requires more than 5 rounds of the iterative process of WDF_NCG algorithm and six instances are solved in 1 round. NTT with threshold 25% requires the largest number of deadlock avoidance constraints, i.e., 11

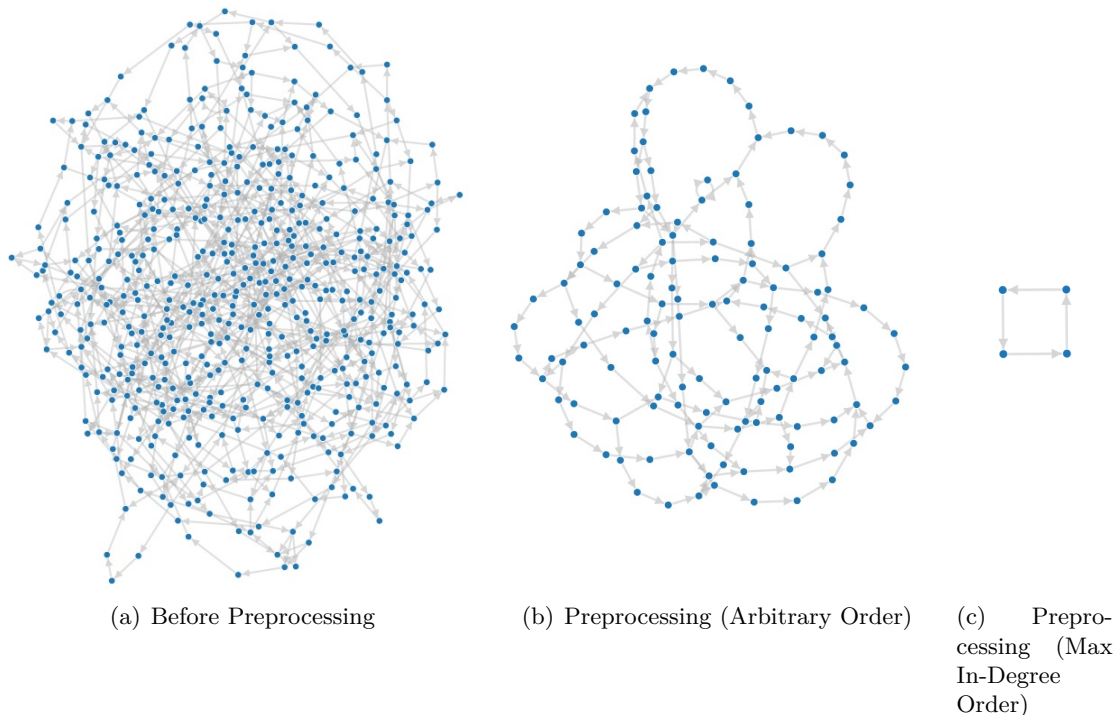


Figure 5.7: Largest SCC for USA 5%

additional constraints.

In addition, `WDF_NCG` algorithm shows to be scalable. All fifteen instances are solved in less than 4 minutes, with computational time of nine instances being less than 1 minute. It should be noted that the time reported in Table 5.6 (column “Time”) measures the time required to calculate the MBB provisioning and does not include the time required for calculating initial RWA^{OPT} that we have reported in column “Min RWA” (time for solving the RWA problem while minimizing the number of the bandwidth).

Table 5.7 provides the bandwidth requirement and how far it is from minimum bandwidth requirement in order to derive a make-before-break reachable wavelength provisioning. We report the bandwidth requirement for the RWA^{FRAG} right before the defragmentation event, the optimal (RWA^{OPT}) provisioning and the best wavelength provisioning (RWA^{MBB_OPT}) that is make-before-break reachable. As Table 5.7 shows, the difference between link usage of the MBB and optimal provisionings is less than 1% and even equal to the optimal provisioning except, NTT (20%) and CONUS (40%) while the performance of

Table 5.6: Performance of the WDF_NCG algorithm

Networks	GoS Reduction	# Rounds	# Constr.	Time (sec)	Min RWA (sec)
NTT	10%	4	4	203	1,259
	15%	1	1	13	1,291
	20%	4	8	168	1,332
	25%	5	11	202	1,153
USA	5%	3	8	70	619
	10%(I)	2	2	28	450
	10%(II)	1	1	11	821
	15%	1	2	15	534
GERMANY	10%	1	2	11	13,158
	15%	2	4	14	18,027
	20%	1	3	10	18,746
CONUS (30)	15%	3	7	115	820
	20%	2	2	52	1,494
	30%	1	2	10	1,561
	40%	2	8	60	1,689
CONUS (50)	25%	3	5	165	2,891
	30%	1	1	83	2,827
	35%	1	2	70	1,069

the network in term of used links is improved in all cases.

Table 5.8 analyzes the total length of the lightpaths in different provisionings based on the geographical distances presented in [108] for the USA network. As expected, the difference between $RWA^{\text{MBB_OPT}}$ and RWA^{OPT} provisionings is negligible while the improvement from RWA^{FRAG} to $RWA^{\text{MBB_OPT}}$ provisioning is over 25%.

Figure 5.8 compares the percentage of the number of the lightpaths routed over the shortest paths in $RWA^{\text{MBB_OPT}}$ and RWA^{FRAG} provisionings. As seen in Figure 5.8, more lightpaths use the shortest paths in $RWA^{\text{MBB_OPT}}$ than in RWA^{FRAG} that was expected based on the higher number of links used in RWA^{FRAG} (Table 5.7). By migrating from RWA^{FRAG} provisioning to MBB provisioning, more lightpaths are routed over shortest paths, fewer links are used and as the result, the blocking rate is reduced.

Table 5.7: Bandwidth requirement compromise for a make-before-break reachable optimized wavelength provisioning

Networks	Defragmentation triggering event	Bandwidth requirement (number of wavelength links)			RWA ^{MBB_OPT}	RWA ^{FRAG}
		RWA ^{FRAG}	RWA ^{OPT}	RWA ^{MBB_OPT}	vs. RWA ^{OPT}	vs. RWA ^{MBB_OPT}
NTT	10%	2,907	2,546	2,558	0.47%	12.00%
	15%	3,009	2,645	2,645	0.00%	12.09%
	20%	2,971	2,537	2,600	2.42%	12.48%
	25%	2,920	2,464	2,470	0.24%	15.41%
USA	5%	3,931	2,798	2,814	0.57%	28.42%
	10%(I)	3,691	2,593	2,601	0.31%	29.53%
	10%(II)	3,690	2,649	2,650	0.04%	28.18%
	15%	3,573	2,545	2,548	0.12%	28.69%
GERMANY	10%	8,347	6,386	6,386	0.00%	23.50%
	15%	7,893	5,984	5,984	0.00%	24.18%
	20%	7,008	5,329	5,329	0.00%	23.96%
CONUS (30)	15%	2,402	1,948	1,948	0.00%	18.90%
	20%	2,308	1,879	1,889	0.53%	18.15%
	30%	2,191	1,575	1,589	0.89%	27.48%
	40%	2,119	1,484	1,508	1.62%	28.83%
CONUS (50)	25%	3,921	3,410	3,424	0.41%	14.51%
	30%	3,746	3,159	3,164	0.15%	15.67%
	35%	3,718	3,039	3,054	0.49%	22.34%

5.6.5 Number of Reroutings vs. Bandwidth Improvement

In this section, we investigate the trade off between the percentage of rerouted lightpaths and bandwidth saving while providing a seamless migration. As illustrated in Figure 5.9, aiming higher bandwidths savings will result in more rerouted lightpaths. Figure 5.9 shows that almost all lightpaths (100%) need to be rerouted in order to reach the maximum bandwidth saving in NTT network (10%) and USA network (15%). The same holds for all

Table 5.8: Length Reduction for USA

Networks	Defragmentation triggering event	Total length of the lightpaths (Length (KM))			RWA ^{MBB_OPT}	RWA ^{FRAG}
		RWA ^{FRAG}	RWA ^{OPT}	RWA ^{MBB_OPT}	vs. RWA ^{OPT}	vs. RWA ^{MBB_OPT}
USA	5%	2,996,850	2,978,500	4,044,700	0.61%	25.90%
	10%(I)	2,761,000	2,751,200	3,709,500	0.35%	25.56%
	10%(II)	2,820,900	2,820,050	3,807,850	0.03%	25.91%
	15%	2,724,200	2,721,450	3,701,550	0.10%	26.40%

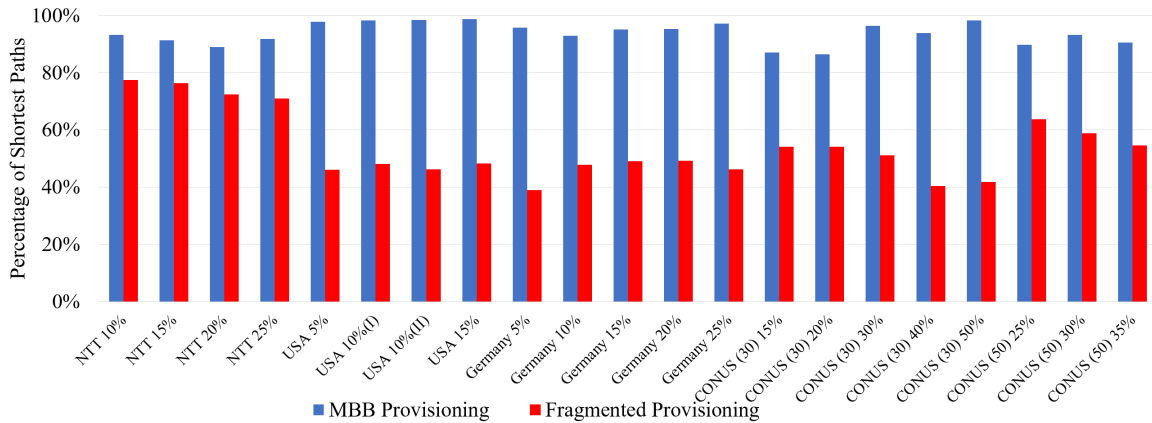
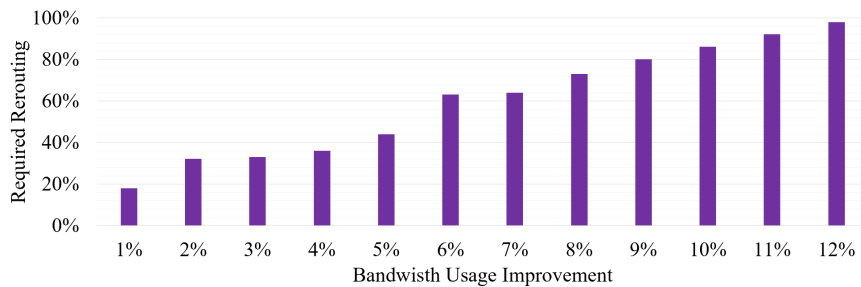


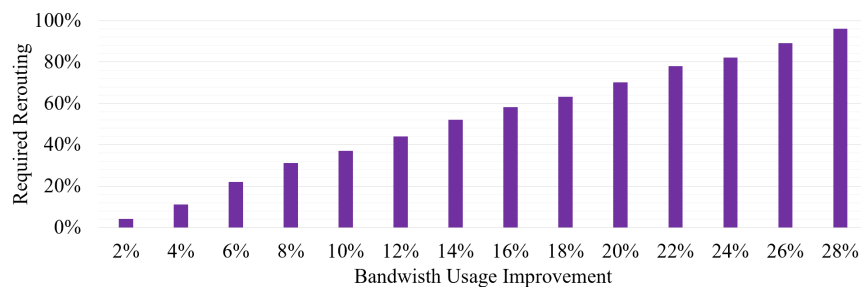
Figure 5.8: Percentage of Shortest Paths

the networks used in this study.

The seamless solution still gives the network operator the opportunity to decide on the desired saving and/or possible amount of reroutings and come up with the best seamless migration. For instance, Figure 5.9(a) shows that if we reroute at most 60% of the lightpaths, then the maximum bandwidth usage will be around 7%.



(a) NTT 10%



(b) USA 15%

Figure 5.9: Rerouting vs. Bandwidth Improvement

5.7 Conclusions

We proposed an efficient algorithm for obtaining an MBB wavelength optimized provisioning that allows a seamless migration from a fragmented network provisioning to an optimized one. We also proposed an algorithm in order to reduce the number of strongly connected components by identifying avoidable disruptions. Although the number and the size of the strongly connected components for migrating from fragmented provisioning to optimal provisioning might be big, applying the proposed algorithm can efficiently reduce both the number and the size of them. This consequently helps us add simpler deadlock avoidance constraints. We evaluated our algorithm on different real size data sets. The results show that the algorithm can efficiently provide a seamless migration in reasonable time.

Chapter 6

Conclusions and Future Work

In this thesis, we study two critical problems in telecommunication industries:

- Network Migration Problem (NMP): providing the plan for upgrading the infrastructure of a network with minimum cost.
- Wavelength Defragmentation Problem: reoptimizing the bandwidth usage by finding a seamless migration for a fragmented optical network.

The list of published papers presenting the results of the thesis are [46, 48, 76, 77] for NMP and [49] for wavelength defragmentation problem. It should be mentioned that we received the best paper award for [77] from SARNOFF Symposium 2017.

In Chapters 3 and 4, we study network migration problem. We prove that the problem is NP-hard and propose two MIP formulations. The model proposed in Chapter 3 is the first try to tackle NMP. We propose a two-phase algorithm based on column generation technique. Extensive numerical experiments are conducted on several real instances with different parameters. The quality of the solutions and the sensitivity of the algorithm toward different parameters are validated by experienced network engineers in Ciena. The second formulation, presented in Chapter 4, takes advantage of removing the symmetry between the circuits. This formulation is capable of providing a strong lower bound for real size instances. We apply a branch-and-price algorithm that shows a promising performance on both artificial and real instances (less than 10% gap between LP and ILP solutions).

In Chapter 5, we propose a methodology in order to find an optimum provisioning that is make-before-break reachable. The methodology is based on nested column generation technique that generates a constraint for every cycle in the dependency graph that hinders us from seamless migration. The algorithm is tested on several real size instances and we can efficiently provide a seamless migration for all.

Future Work

Both models provided in Chapters 3 and 4, formulate NMP under the assumption that the new network is already available and all circuits are eligible to be migrated in all maintenance windows. Future work includes the study of migrating a legacy network along with deploying the new network. In this case, the order of deploying the new equipment in the sites will be a decision variable that will affect the order and availability of circuits for migration.

As mentioned in Chapter 4, NMP is a variant of VRPS. The proposed methodology in Chapter 4 for NMP can efficiently handle real size instances that are much bigger than the instances solved in VRPS. One of the future research directions is to investigate a general framework for VRPS based on the proposed methodology.

The solution method presented in Chapter 5 does not take into account the number of reroutings. Future work should consider investigating the best compromise between number of reroutings and bandwidth savings, as well as the triggering of the defragmentation events.

Bibliography

- [1] Alcatel-Lucent. Transformation of mission-critical communications networks migrating from SDH/SONET networks to IP/MPLS networks. Technology White Paper, 2015.
- [2] A. A. Almughales and A. M. Alsaih. Optimum migration scenario from pstn to ngn. In *IEEE International Conference on Communication Systems, Networks and Applications (ICCSNA)*, pages 227–231, 2010.
- [3] Kapil Bakshi. Considerations for software defined networking (SDN): Approaches and use cases. In *IEEE Aerospace Conference*, pages 1–9, 2013.
- [4] C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, and P.H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329, 1998.
- [5] Cynthia Barnhart, Ellis L Johnson, George L Nemhauser, Martin WP Savelsbergh, and Pamela H Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations research*, 46(3):316–329, 1998.
- [6] M. Batayneh, D.A. Schupke, M. Hoffmann, A. KIRSTAEDTER, and B. Mukherjee. On routing and transmission-range determination of multi-bit-rate signals over mixed-line-rate WDM optical networks for carrier ethernet. *IEEE/ACM Transactions on Networking*, 19:1304–1316, October 2011.
- [7] R. J. Bates. Sonet and sdh. In *Broadband Telecommunications Handbook*, chapter 27, pages 451–473. McGraw-Hill, 2000.

- [8] Sonia Belhareth, David Coudert, Dorian Mazauric, Nicolas Nisse, and Issam Tahiri. Reconfiguration with physical constraints in WDM networks. In *IEEE International Conference on Communications - ICC*, pages 6257–6261, 2012.
- [9] Partha Bhaumik, Shuqiang Zhang, Pulak Chowdhury, Sang-Soo Lee, Jong Hyun Lee, and Biswanath Mukherjee. Software-defined optical networks (SDONs): a survey. *Photonic Network Communications*, 28(1):4–18, 2014.
- [10] A. Bley, F. D’Andreagiovanni, and D. Karch. Scheduling technology migration in WDM networks. In *ITG Symposium on Photonic Networks*, pages 1–5, 2013.
- [11] D. Bredström and M. Rönnqvist. Combined vehicle routing and scheduling with temporal precedence and synchronization constraints. *European Journal of Operational Research*, 191:19–31, November 2008.
- [12] David Bredstrom and Mikael Rönnqvist. A branch and price algorithm for the combined vehicle routing and scheduling problem with synchronization constraints. Technical report, Department of Finance and Management Science, Norwegian School of Economics and Business Administration, 2007.
- [13] J.A. Castillo-Salazar, D. Landa-Silva, and R. Qu. Workforce scheduling and routing problems: literature survey and computational study. *Annals of Operations Research*, 239(1):39–67, 2016.
- [14] Chien Chen and Subrata Banerjee. A new model for optimal routing and wavelength assignment in wavelength division multiplexed optical networks. In *IEEE Annual Joint Conference of the IEEE Computer and Communications Societies - INFOCOM*, volume 1, pages 164–171, 1996.
- [15] Imrich Chlamtac, Aura Ganz, and Gadi Karmi. Lightpath communications: An approach to high bandwidth optical WAN’s. *IEEE transactions on communications*, 40(7):1171–1182, 1992.
- [16] V. Chvatal. *Linear Programming*. Freeman, 1983.

- [17] Ciena. SONET/SDH network modernization is long overdue, 2013.
- [18] Cisco. How Cisco IT migrated TDM local access from SONET to OC-192 infrastructure. Cisco IT Case Study, 2007.
- [19] CISCO. *Cisco Visual Networking Index: Forecast and Methodology, 2016–2021*, June 2017.
- [20] Nathann Cohen, David Coudert, Dorian Mazaauric, Napoleao Nepomuceno, and Nicolas Nisse. Tradeoffs in process strategy games with application in the WDM reconfiguration problem. *Theoretical Computer Science*, 412(35):4675–4687, 2011.
- [21] C. Cooper and A. Frieze. The size of the largest strongly connected component of a random digraph with a given degree sequence. *Combinatorics, Probability and Computing*, 13(3):319 – 337, 2004.
- [22] Coriant. Intelligent modernization and migration. Coriant White Paper, 2014.
- [23] Coriant. Telstra migrates to IP/MPLS network core in under 18 months. Coriant White Paper, 2014.
- [24] David Coudert, Florian Huc, Dorian Mazaauric, Nicolas Nisse, and Jean-Sébastien Sereni. Reconfiguration of the routing in WDM networks with two classes of services. In *Conference on Optical Network Design and Modeling - ONDM*, pages 1–6, 2009.
- [25] David Coudert, Stéphane Pérennes, Quang-Cuong Pham, and Jean-Sébastien Sereni. Rerouting requests in WDM networks. In *7eme Rencontres Francophones sur les aspects Algorithmiques des Télécommunications - AlgoTel*, pages 17–20, 2005.
- [26] David Coudert and Hervé Rivano. Lightpath assignment for multifibers WDM networks with wavelength translators. In *IEEE Global Telecommunications Conference - GLOBECOM*, volume 3, pages 2686–2690, Taipei, Taiwan, 2002. IEEE.
- [27] David Coudert and Jean-Sébastien Sereni. Characterization of graphs and digraphs with small process number. *Discrete Applied Mathematics*, 159(11):1094–1109, July 2011.

- [28] G. Desaulniers, J. Desrosiers, and M. M. Solomon, editors. *Column Generation*. GERAD 25th Anniversary Series. Springer, 2005.
- [29] M. Drexl. Synchronization in vehicle routing—a survey of VRPs with multiple synchronization constraints. *Transportation Science*, 46(3):297–316, 2012.
- [30] Michael Drexl. Synchronization in vehicle routing—a survey of VRPs with multiple synchronization constraints. *Transportation Science*, 46(3):297–316, 2012.
- [31] C. Duhamel, P. Mahey, A.X. Martins, R.R. Saldanha, and M.C. de Souza. Model-hierarchical column generation and heuristic for the routing and wavelength assignment problem. *4OR*, pages 1 – 20, 2016.
- [32] Christiane Dzungang, Philippe Galinier, and Samuel Pierre. A Tabu Search heuristic for the routing and wavelength assignment problem in optical networks. *IEEE Communications letters*, 9(5):426–428, 2005.
- [33] Burak Eksioglu, Arif Volkan Vural, and Arnold Reisman. The vehicle routing problem: A taxonomic review. *Computers & Industrial Engineering*, 57(4):1472–1483, 2009.
- [34] V. Eramo, E. Miucci, M. Ammar, and F.G. Lavacca. An approach for service function chain routing and virtual function network instance migration in network function virtualization architectures. *IEEE/ACM Transactions on Networking*, 25:2008–2025, 2017.
- [35] Wissam Fawaz, Belkacem Daheb, Olivier Audouin, Michel Du-Pond, and Guy Pujolle. Service level agreement and provisioning in optical networks. *IEEE Communications Magazine*, 42(1):36–43, 2004.
- [36] A. B. Forouzan. *Data Communications & Networking*. Tata McGraw-Hill Education, 2006.
- [37] Fujitsu. Packet optical transport: Migrating networks with next gen packet optical network platforms. European Conference and Exhibition on Optical Communication (ECOC) Exhibition, 2010.

- [38] Open Networking Foundation. Software-defined networking: The new norm for networks. ONF White Paper, 2012.
- [39] Harold N Gabow. Path-based depth-first search for strong and biconnected components. *Information Processing Letters*, 74(3-4):107–114, 2000.
- [40] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. A Series of books in the mathematical sciences. W. H. Freeman, 1979.
- [41] W. Golab and R. Boutaba. Policy-driven automated reconfiguration for performance management in WDM optical networks. *IEEE Communications Magazine*, 42(1):44 – 51, 2003.
- [42] M. Jaeger and R. Huelsermann. Network migration cost study. In *Conference on Optical Network Design and Modeling - ONDM*, pages 1–6, 2010.
- [43] B. Jaumard and M. Daryalal. Efficient spectrum utilization in large scale rwa problems. *IEEE/ACM Transactions on Networking*, PP:1–16, March 2017.
- [44] B. Jaumard, C. Meyer, and B. Thiongane. ILP formulations for the RWA problem for symmetrical systems. In P. Pardalos and M. Resende, editors, *Handbook for Optimization in Telecommunications*, chapter 23, pages 637–678. Kluwer, 2006.
- [45] B. Jaumard, C. Meyer, and B. Thiongane. On column generation formulations for the RWA problem. *Discrete Applied Mathematics*, 157:1291–1308, March 2009.
- [46] B. Jaumard, H. Pouya, R. Fahim, and A. Barrios. Planning network migration. In *IEEE International Conference on Communications - ICC*, pages 1–6, 2016.
- [47] Brigitte Jaumard, Christophe Meyer, and Babacar Thiongane. Comparison of ILP formulations for the RWA problem. *Optical Switching and Networking*, 4(3):157–172, 2007.
- [48] Brigitte Jaumard and Hamed Pouya. Migration plan with minimum overall migration time or cost. *Journal of Optical Communications and Networking*, 10(1):1–13, 2018.

- [49] Brigitte Jaumard, Hamed Pouya, and David Coudert. Make-before-break wavelength defragmentation. In *IEEE International Conference on Transparent Optical Networks - ICTON*, pages 1–5, 2018.
- [50] Brigitte Jaumard, Hamed Pouya, Rami Fahim, and Andres Barrios. Planning network migration. In *IEEE International Conference on Communications - ICC*, pages 1–6, 2016.
- [51] Ellis L Johnson. Modeling and strong linear programs for mixed integer programming. In *Algorithms and model formulations in mathematical programming*, pages 1–43. Springer, 1989.
- [52] N. Jose and K. Somani. Connection rerouting/network reconfiguration. In *IEEE Conference on Design of Reliable Communication Networks - DRCN*, pages 23 – 30, Banff, Alberta, Canada, October 2003.
- [53] Nitin Jose and Arun K Somani. Connection rerouting/network reconfiguration. In *IEEE Conference on Design of Reliable Communication Networks - DRCN*, pages 23–30, 2003.
- [54] B. G. Józsa and M. Makai. On the solution of reroute sequence planning problem in MPLS networks. *Computer Networks*, 42(2):199 – 210, 2003.
- [55] Akihiro Kadohata, Akira Hirano, Mitsunori Fukutoku, Takuya Ohara, Yoshiaki Sone, and Osamu Ishida. Multi-layer greenfield re-grooming with wavelength defragmentation. *IEEE communications letters*, 16(4):530–532, 2012.
- [56] O. Klopfenstein. Rerouting tunnels for MPLS network resource optimization. *European Journal of Operational Research*, 188(1):293 – 312, 2008.
- [57] M Shiva Kumar and P Sreenivasa Kumar. Static lightpath establishment in WDM networks—new ILP formulations and heuristic algorithms. *Computer Communications*, 25(1):109–114, 2002.

- [58] Nacima Labadie, Christian Prins, and Yanyan Yang. Iterated local search for a vehicle routing problem with synchronization constraints. In *International Conference on Operations Research and Enterprise Systems - ICORES*, pages 257–263, 2014.
- [59] K.-C. Lee and V. Li. A wavelength rerouting algorithm in wide-area all-optical networks. *Journal of Lightwave Technology*, 14(6):1218–1229, 1996.
- [60] Kyungsik Lee, Kug Chang Kang, Taehan Lee, and Sungsoo Park. An optimization approach to routing and wavelength assignment in WDM all-optical mesh networks without wavelength conversion. *ETRI journal*, 24(2):131–141, 2002.
- [61] A. Leiva, C.M. Machuca, A. Beghelli, and R. Olivares. Migration cost analysis for upgrading WDM networks. *IEEE Communications Magazine*, pages 87–93, Nov. 2013.
- [62] C.-L. Li, S.T. McCormick, and D. Simchi-Levi. Finding disjoint paths with different path-costs: Complexity and algorithms. *Networks*, 22:653–67, 1992.
- [63] H. Li and J. Wu. Survey of WDM network reconfiguration: topology migrations and their impact on service disruptions. *Telecommunications Systems*, 60:349–366, 2015.
- [64] P. Littlewood and Masoud F. *Optical Transport Networking*. Ciena, 2014.
- [65] Luciano Margara and Janos Simon. Wavelength assignment problem on all-optical networks with k fibres per link. In *International Colloquium on Automata, Languages, and Programming*, pages 768–779. Springer, 2000.
- [66] A.X. Martins, C. Duhamel, P. Mahey, R.R. Saldanh, and Mauricio C. de Souza. Variable neighborhood descent with iterated local search for routing and wavelength assignment. *Computers & Operations Research*, 39:2133–2141, 2012.
- [67] M. Melo, S. Sargento, and J. Carapinha. Optimal virtual network migration: A step closer for seamless resource mobility. *Journal of Network and Computer Applications*, 64:124–136, Apr. 2016.

- [68] Takashi Miyamura, Eiji Oki, Ichiro Inoue, and Kohei Shiimoto. Enhancing bandwidth on demand service based on virtual network topology control. In *IEEE Network Operations and Management Symposium - NOMS*, pages 201–206, 2008.
- [69] Gurusamy Mohan and C Siva Ram Murthy. A time optimal wavelength rerouting algorithm for dynamic traffic in WDM networks. *Journal of lightwave technology*, 17(3):406, 1999.
- [70] Son Hong Ngo, Xiaohong Jiang, and Susumu Horiguchi. An ant-based approach for dynamic RWA in optical WDM networks. *Photonic Network Communications*, 11(1):39–48, 2006.
- [71] Bruno Astuto A Nunes, Marc Mendonca, Xuan-Nam Nguyen, Katia Obraczka, and Thierry Turletti. A survey of software-defined networking: Past, present, and future of programmable networks. *IEEE Communications Surveys & Tutorials*, 16(3):1617–1634, 2014.
- [72] S. Orłowski, M. Pióro, A. Tomaszewski, and R. Wessäly. SNDlib 1.0—Survivable Network Design Library. In *International Network Optimization Conference - INOC*, pages 276–286, April 2007.
- [73] Francesco Palmieri, Ugo Fiore, and Sergio Ricciardi. A GRASP-based network re-optimization strategy for improving RWA in multi-constrained optical transport infrastructures. *Computer Communications*, 33(15):1809–1822, 2010.
- [74] P. Podhradsky. Migration scenarios and convergence processes towards NGN (present state and future trends). In *46th IEEE International Symposium Electronics in Marine*, pages 39–46, 2004.
- [75] Pavol Podhradsky. Migration scenarios and convergence processes towards NGN (present state and future trends). In *IEEE International Symposium Electronics in Marine*, pages 39–46, 2004.

- [76] Hamed Pouya and Brigitte Jaumard. Efficient network migration planning. In *IEEE International Conference on Transparent Optical Networks - ICTON*, pages 1–4, 2017.
- [77] Hamed Pouya, Brigitte Jaumard, and Catherine Preston-Thomas. Minimum network migration cost and duration. In *IEEE Sarnoff Symposium*, pages 1–6, 2017.
- [78] R. Ramaswami, K. Sivarajan, and G. Sasaki. *Optical networks: a practical perspective*. Morgan Kaufmann, 2009.
- [79] Rajiv Ramaswami and Kumar N Sivarajan. Routing and wavelength assignment in all-optical networks. *IEEE/ACM Transactions on Networking*, 3(5):489–500, 1995.
- [80] Line Blander Reinhardt, Tommy Clausen, and David Pisinger. Synchronized dial-a-ride transportation of disabled passengers at airports. *European Journal of Operational Research*, 225(1):106–117, 2013.
- [81] Shiva Rowshanrad, Sahar Namvarasl, Vajihe Abdi, Maryam Hajizadeh, and Manijeh Keshtgary. A survey on sdn, the future of networking. *Journal of Advanced Computer Science & Technology*, 3(2):232–248, 2014.
- [82] M Angélica Salazar-Aguilar, André Langevin, and Gilbert Laporte. Synchronized arc routing for snow plowing operations. *Computers & Operations Research*, 39(7):1432–1440, 2012.
- [83] M Angélica Salazar-Aguilar, André Langevin, and Gilbert Laporte. The synchronized arc and node routing problem: Application to road marking. *Computers & Operations Research*, 40(7):1708–1715, 2013.
- [84] M.A. Salazar-Aguilar, A. Langevin, and G. Laporte. Synchronized arc routing for snow plowing operations. *Computers & Operations Research*, 39:1432 – 1440, July 2012.
- [85] Martin Savelsbergh. A branch-and-price algorithm for the generalized assignment problem. *Operations research*, 45(6):831–841, 1997.

- [86] Micha Sharir. A strong-connectivity algorithm and its applications in data flow analysis. *Computers & Mathematics with Applications*, 7(1):67–72, 1981.
- [87] J. M. Simmons. *Optical network design and planning*. Springer, 2014.
- [88] J. Simons. CONUS network. Monarch Network Architects, <http://www.monarchna.com/topology.html>.
- [89] Fernando Solano. Analyzing two different objectives of the WDM lightpath reconfiguration problem. In *IEEE Global Telecommunications Conference - GLOBECOM*, pages 6491–6497, 2009.
- [90] N Sreenath and C Siva Ram Murthy. On-line reconfiguration of virtual topologies in wavelength-routed WDM networks. *Journal of High Speed Networks*, 12(3, 4):141–169, 2002.
- [91] MD Swaminathan and Kumar N Sivarajan. Practical routing and wavelength assignment algorithms for all optical networks with limited wavelength conversion. In *IEEE International Conference on Communications - ICC*, volume 5, pages 2750–2755, 2002.
- [92] Y. Takita, K. Tajima, T. Hashiguchi, and T. Katagiri. Wavelength defragmentation for seamless service migration. *Journal of Optical Communications and Networking*, 9(2):A154–A161, 2017.
- [93] R.E. Tarjan. Depth-first search and linear graph algorithms. *SIAM journal on computing*, 1(2):146–160, 1972.
- [94] Akhilesh S Thyagaturu, Anu Mercian, Michael P McGarry, Martin Reisslein, and Wolfgang Kellerer. Software defined optical networks (SDONs): A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 18(4):2738–2786, 2016.
- [95] S. Türk, Y. Liu, R. Radeke, and R. Lehnert. Network migration optimization using genetic algorithms. In *Information and Communication Technologies*, pages 112–123, 2012.

- [96] S. Türk, S. Sulaiman, A. Haidine, R. Lehnert, and T. Michaelis. Approaches for the migration of optical backbone networks towards carrier Ethernet. In *IEEE Global Telecommunications Conference - GLOBECOM*, pages 1–6, 2009.
- [97] Pamela H Vance. Branch-and-price algorithms for the one-dimensional cutting stock problem. *Computational optimization and applications*, 9(3):211–228, 1998.
- [98] François Vanderbeck. Computational study of a column generation algorithm for bin packing and cutting stock problems. *Mathematical Programming*, 86(3):565–594, 1999.
- [99] François Vanderbeck. Branching in branch-and-price: a generic scheme. *Mathematical Programming*, 130(2):249–294, 2011.
- [100] François Vanderbeck and Laurence A Wolsey. An exact algorithm for IP column generation. *Oper. Res. Letters*, 19(4):151–159, 1996.
- [101] M.A. Vega-Rodriguez and A. Rubio-Largo. NTT network. http://mstar.unex.es/mstar_documentos/RWA/RWA-Instances.html.
- [102] John Y Wei. Advances in the management and control of optical internet. *IEEE Journal on Selected Areas in Communications*, 20(4):768–785, 2002.
- [103] J. Wu. A survey of WDM network reconfiguration: strategies and triggering method. *Computer Networks*, 55(11):2622–2645, 2011.
- [104] Yawei Yin, Ke Wen, David J Geisler, Ruiting Liu, and SJB Yoo. Dynamic on-demand defragmentation in flexible bandwidth elastic optical networks. *Optics express*, 20(2):1798–1804, 2012.
- [105] Mingyang Zhang, Changsheng You, Huihui Jiang, and Zuqing Zhu. Dynamic and adaptive bandwidth defragmentation in spectrum-sliced elastic optical networks with time-varying traffic. *Journal of Lightwave Technology*, 32(5):1014–1023, 2014.

- [106] Mingyang Zhang, Changsheng You, and Zuqing Zhu. On the parallelization of spectrum defragmentation reconfigurations in elastic optical networks. *IEEE/ACM Transactions on Networking*, 24(5):2819–2833, 2016.
- [107] Yongbing Zhang, Masaki Murata, Hideaki Takagi, and Yusheng Ji. Traffic-based reconfiguration for logical topologies in large-scale WDM optical networks. *Journal of lightwave technology*, 23(10):2854, 2005.
- [108] Zuqing Zhu, Wei Lu, Liang Zhang, and Nirwan Ansari. Dynamic service provisioning in elastic optical networks with hybrid single-/multi-path routing. *Journal of Lightwave Technology*, 31(1):15–22, 2013.

Appendix

Appendix .A Generic Pricing Problem (GPP) for Model in Section 4.4.2

The ILP formulation presented in this section is the generic pricing problem for the formulation in Section 4.4.2. The decision variable p_t in the following model is equal to 1 if technician $t \in T$ works in plan p , 0 otherwise. T_r represents the set of the technicians that belong to region $r \in R$. The rest of the variables and parameters are the same as the ones introduced in Section 4.3 except that index t representing a technician is added to some.

$$[\text{GPP}_w] \quad \min \quad \text{cost} - \pi_w^{(4.33)} - \sum_{s \in S_p} \sum_{s' \in S_p} m_{ss'} \pi_{ss'}^{(4.34)} \quad (.1)$$

subject to:

$$\text{cost} = \sum_{t \in T} (\text{COST}^{\text{TECH}} + \text{COST}^{\text{ENG}}) \Delta_{\text{SHIFT}}^t \quad (.2)$$

$$\sum_{t \in T_r} p_t \leq \bar{n}_r^{\text{TECH}} \quad r \in R \quad (.3)$$

$$\frac{1}{\alpha^{\text{ENG}}} \sum_{t \in T} p_t \leq \bar{n}^{\text{ENG}} \quad (.4)$$

$$\sum_{t \in T} n_{\text{CIR}}^t \leq 2\bar{n}^{\text{CIR}} \quad (.5)$$

$$n_{\text{CIR}}^t = \sum_{s \in S_r} \sum_{s' \in S} n_{ss'}^t \quad r \in R, t \in T_r \quad (.6)$$

$$\sum_{t \in T} n_{ss'}^t = l_{ss'} \quad s, s' \in S_p \quad (.7)$$

$$m_{ss'} \leq l_{ss'} \quad s, s' \in S_p, s < s' \quad (.8)$$

$$m_{ss'} \leq l_{s's} \quad s, s' \in S_p, s < s' \quad (.9)$$

$$\sum_{s \in S_r} \sum_{s' \in S} n_{ss'}^t \leq Mp_t \quad r \in R, t \in T_r \quad (.10)$$

$$h_{ss'}^t + h_{s's}^t \leq 1 \quad r \in R, t \in T_r, s, s' \in S_r, s \neq s' \quad (.11)$$

$$h_s^t \leq \sum_{s' \in S} h_{ss'}^t \leq Mh_s^t \quad r \in R, t \in T_r, s \in S_r \quad (.12)$$

$$h_{ss'}^t \leq n_{ss'}^t \leq Mh_{ss'}^t \quad r \in R, t \in T_r, s \in S_r, s' \in S \quad (.13)$$

$$\sum_{s \in S_r} t_{\text{SSRC},s}^t = p_t, \quad \sum_{s \in S_r} t_{s,\text{SDST}}^t = p_t \quad r \in R, t \in T_r \quad (.14)$$

$$t_{ss'}^t \leq \frac{1}{2}(h_s^t + h_{s'}^t) \quad r \in R, t \in T_r, s, s' > s \in S_r \quad (.15)$$

$$\sum_{\substack{s' \in S_r^+ \\ s' > s}} t_{ss'}^t + \sum_{\substack{s' \in S_r^+ \\ s' < s}} t_{s's}^t = 2h_s^t \quad r \in R, t \in T_r, s \in S_r \quad (.16)$$

$$\sum_{s \in S_r} \sum_{\substack{s' \in S_r \\ s' > s}} t_{ss'}^t = \sum_{s \in S_r} h_s^t - p_t \quad r \in R, t \in T_r \quad (.17)$$

$$\bar{\Delta}_{\text{MIGR}} \sum_{s \in S_r} \sum_{\substack{s' \in S_r \\ s \neq s' \\ s' > s}} n_{ss'}^t + \sum_{s \in S_r} \sum_{\substack{s' \in S_r \\ s' > s}} T_{ss'} t_{ss'}^t \leq \Delta_{\text{SHIFT}}^t \quad r \in R, t \in T_r \quad (.18)$$

$$\sum_{\delta \in \bar{\Delta}_{\text{SHIFT}}} x_{\delta}^t = p_t \quad t \in T \quad (.19)$$

$$\sum_{\delta \in \bar{\Delta}_{\text{SHIFT}}} \bar{\Delta}_{\delta} x_{\delta}^t = \Delta_{\text{SHIFT}}^t \quad t \in T \quad (.20)$$

Constraint set (.2) calculates the cost of plan p . Constraint sets (.3) - (.5) are equivalent to constraint sets (4.4) - (4.6) for maintenance window w . Constraint set (.6) is equivalent to constraint (4.15). Constraint sets (.7) - (.9) have the same functionality as the constraint sets (4.8) - (4.10). Constraint set (.10) determines which technicians will work in plan p . Constraint sets (.11) - (.20) are equivalent to constraint sets (4.16) - (4.25).

Appendix .B Pricing Problem (PrPP) Using a Set of Pre-computed Shifts for Model in Section 4.4.2

The following formulation generates a plan for a given maintenance window w based on Γ_w , a set of precomputed shifts for w . The decision variable y_γ determines the number of times shift γ is selected in the current plan.

$$[\text{PrPP}_w] \quad \min \quad \text{cost} - \pi_w^{(4.33)} - \sum_{s \in S_p} \sum_{s' \in S_p} m_{ss'} \pi_{ss'}^{(4.34)} \quad (.21)$$

subject to:

$$\text{cost} = \sum_{\gamma \in \Gamma_w} (\text{COST}^{\text{TECH}} + \text{COST}^{\text{ENG}}) \Delta_{\text{SHIFT}}^\gamma y_\gamma \quad (.22)$$

$$\sum_{\gamma \in \Gamma_w} n_{ss'}^\gamma y_\gamma = l_{ss'} \quad s, s' \in S \quad (.23)$$

$$m_{ss'} \leq l_{ss'} \quad s, s' \in S_p, s < s' \quad (.24)$$

$$m_{ss'} \leq l_{s's} \quad s, s' \in S_p, s < s' \quad (.25)$$

$$\sum_{\gamma \in \Gamma_{rw}} y_\gamma \leq \bar{n}_r^{\text{TECH}} \quad r \in R \quad (.26)$$

$$\frac{1}{\alpha^{\text{ENG}}} \sum_{\gamma \in \Gamma_w} y_\gamma \leq \bar{n}^{\text{ENG}} \quad (.27)$$

$$\sum_{\gamma \in \Gamma_w} n_{\text{CIR}}^\gamma y_\gamma \leq 2\bar{n}^{\text{CIR}} \quad (.28)$$

Constraint set (.22) calculates the migration costs for the current plan. Constraint sets (.23)-(.25) have the same functionality as the constraint sets (4.8) - (4.10). Constraint sets (.26) - (.28) are equivalent to the constraint sets (4.4) - (4.6) for maintenance window w .