Université de Montréal

Optimization of p-cycle protection schemes in optical networks

par
Caroline Thennecy de Medeiros Rocha

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Thèse présentée à la Faculté des études supérieures et postdoctorales
en vue de l'obtention du grade de Philosophiæ Doctor (Ph.D.)
en informatique

Août, 2009

Université de Montréal
Faculté des études supérieures et postdoctorales

Cette thèse intitulée:

Optimization of p-cycle protection schemes in optical networks

présentée par:

Caroline Thennecy de Medeiros Rocha

a été évaluée par un jury composé des personnes suivantes:

Jacques Ferland,	président-rapporteur
Brigitte Jaumard,	directeur de recherche
Michel Gendreau,	membre du jury
Arie Koster,	examinateur externe
Claude Carignan,	représentant du doyen de la FESP

Thèse acceptée le: 07 octobre 2009

RÉSUMÉ

La survie des réseaux est un domaine d'étude technique très intéressant ainsi qu'une préoccupation critique dans la conception des réseaux. Compte tenu du fait que de plus en plus de données sont transportées à travers des réseaux de communication, une simple panne peut interrompre des millions d'utilisateurs et engendrer des millions de dollars de pertes de revenu. Les techniques de protection des réseaux consistent à fournir une capacité supplémentaire dans un réseau et à réacheminer les flux automatiquement autour de la panne en utilisant cette disponibilité de capacité.

Cette thèse porte sur la conception de réseaux optiques intégrant des techniques de survie qui utilisent des schémas de protection basés sur les p-cycles. Plus précisément, les p-cycles de protection par chemin sont exploités dans le contexte de pannes sur les liens. Notre étude se concentre sur la mise en place de structures de protection par p-cycles, et ce, en supposant que les chemins d'opération pour l'ensemble des requêtes sont définis a priori. La majorité des travaux existants utilisent des heuristiques ou des méthodes de résolution ayant de la difficulté à résoudre des instances de grande taille. L'objectif de cette thèse est double. D'une part, nous proposons des modèles et des méthodes de résolution capables d'aborder des problèmes de plus grande taille que ceux déjà présentés dans la littérature. D'autre part, grâce aux nouveaux algorithmes, nous sommes en mesure de produire des solutions optimales ou quasi-optimales. Pour ce faire, nous nous appuyons sur la technique de génération de colonnes, celle-ci étant adéquate pour résoudre des problèmes de programmation linéaire de grande taille. Dans ce projet, la génération de colonnes est utilisée comme une façon intelligente d'énumérer implicitement des cycles prometteurs.

Nous proposons d'abord des formulations pour le problème maître et le problème auxiliaire ainsi qu'un premier algorithme de génération de colonnes pour la conception de réseaux protegées par des p-cycles de la protection par chemin. L'algorithme obtient de meilleures solutions, dans un temps raisonnable, que celles obtenues par les méthodes existantes. Par la suite, une formulation plus compacte est proposée pour le problème auxiliaire. De plus, nous présentons une nouvelle méthode de décomposition hiérarchique qui apporte une grande amélioration de l'efficacité globale de l'algorithme. En ce qui concerne les solutions en nombres entiers, nous proposons deux méthodes heurisiques qui arrivent à trouver des bonnes solutions.

Nous nous attardons aussi à une comparaison systématique entre les p-cycles et les schémas classiques de protection partagée. Nous effectuons donc une comparaison précise en utilisant des formulations unifiées et basées sur la génération de colonnes pour obtenir des résultats de bonne qualité. Par la suite, nous évaluons empiriquement les versions orientée et non-orientée des p-cycles pour la protection par lien ainsi que pour la protection par chemin, dans des scénarios de trafic asymétrique. Nous montrons quel est le coût de protection additionnel engendré lorsque des systèmes bidirectionnels sont employés dans de tels scénarios.

Finalement, nous étudions une formulation de génération de colonnes pour la conception de réseaux avec des p-cycles en présence d'exigences de disponibilité et nous obtenons des premières bornes inférieures pour ce problème.

Mots clés : Conception de réseaux, Télécommunications, Protection partagée, Génération de colonnes

ABSTRACT

Network survivability is a very interesting area of technical study and a critical concern in network design. As more and more data are carried over communication networks, a single outage can disrupt millions of users and result in millions of dollars of lost revenue. Survivability techniques involve providing some redundant capacity within the network and automatically rerouting traffic around the failure using this redundant capacity.

This thesis concerns the design of survivable optical networks using p-cycle based schemes, more particularly, path-protecting p-cycles, in link failure scenarios. Our study focuses on the placement of p-cycle protection structures assuming that the working routes for the set of connection requests are defined *a priori*. Most existing work carried out on p-cycles concerns heuristic algorithms or methods suffering from critical lack of scalability. Thus, the objective of this thesis is twofold: on the one hand, to propose scalable models and solution methods enabling to approach larger problem instances and on the other hand, to produce optimal or near optimal solutions with mathematically proven optimality gaps. For this, we rely on the column generation technique which is suitable to solve large scale linear programming problems. Here, column generation is used as an intelligent way of implicitly enumerating promising cycles to be part of p-cycle designs.

At first, we propose mathematical formulations for the master and the pricing problems as well as the first column generation algorithm for the design of survivable networks based on path-protecting p-cycles. The resulting algorithm obtains better solutions within reasonable running time in comparison with existing methods. Then, a much more compact formulation of the pricing problem is obtained. In addition, we also propose a new hierarchical decomposition method which greatly improves the efficiency of the whole algorithm and allows us to solve larger problem instances. As for integer solutions, two heuristic approaches are proposed to obtain good solutions.

Next, we dedicate our attention to a systematic comparison of p-cycles and classical shared protection schemes. We perform an accurate comparison by using a unified col-

umn generation framework to find provably good results. Afterwards, our study concerns an empirical evaluation of directed and undirected link- and path-protecting p-cycles under asymmetric traffic scenarios. We show how much additional protection cost results from employing bidirectional systems in such scenarios.

Finally, we investigate a column generation formulation for the design of p-cycle networks under availability requirements and obtain the first lower bounds for the problem.

Keywords: Network design, Telecommunications, Shared protection, Column generation.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

To my loving husband and best friend, Daniel.

ACKNOWLEDGMENTS

First and foremost, I offer my gratitude to my supervisor, Professor Brigitte Jaumard, who supported me throughout my thesis with her encouragement and knowledge. This work would not have been possible without her effort, patience and avalaibiliy during these four years, specially in the final stages. I also would like to thank all members of the thesis committee for their prompt evaluation and comments.

All my research work was financially supported by the Brazilian National Council for Scientific and Technological Development (CNPq) and I am extremely grateful to every single Brazilian citizen for her/his precious contribution.

The Department of Computer Science and Operations Research (DIRO) and the Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) at Université de Montréal provided the support and equipment I needed to produce and complete my thesis. I thank the technical and administrative staff members for being abundantly helpful and having assisted me in numerous ways.

I also would like to thank my Quebecker and “almost" Quebecker friends, specially Marie-Ève, Julie, Simon, Nadia, Benoit, Catherine, Claudio, and Gerardo, for sharing with me their culture and showing me many different worlds. They brought me light and warmth when winter was not so friendly. In addition, I thank my office colleagues for the pleasant environment they have provided me with.

Meeting my Brazilian friends, Alysson, Gisela, Patricia, and Rafael, was undoubtably one of the best things that happened to me in these last years. I will forever be grateful to them for enjoying with me many memorable moments and for being so supportive and thoughtful in the difficult ones.

I thank my dear parents, Lourenço and Fátima, and my sisters and brother, Lourena, Segundo, and Aline, on whose unconditional encouragement and love I relied throughout my entire time at the Academy.

I could not end without thanking my husband, Daniel, who tirelessly supported and loved me, even when I was being unloveable. Without him I could not have made it this far.

CHAPTER 1

INTRODUCTION

Providing resilience against failures is an important requirement for many high speed networks. As these networks carry more and more data, a single outage can disrupt millions of users and result in millions of dollars of lost revenue. Thus, *network survivability* becomes a critical concern in network design and in its real-time operation [38].

A connection traverses several nodes in the network between its source and its destination, and there are several elements along its path that can fail. In most cases, failures occur due to natural disasters, such as power outages, fires, or earthquakes, human errors, such as accidental cable cuts, or wrong operation of a switch, and failure of active components inside network equipment, such as transmitters, receivers or controllers [125].

Network failures commonly arise in the form of link failures and node failures. Link failures are mostly caused by fiber cable cut, which is the most likely failure event. For instance, U.S. carriers reported 136 such failures to the Federal Communications Commission (FCC) in 1997 [95]. Fiber cuts often result from accidental or malicious excavation, or inline component disfunction such as the failure of an optical amplifier. A node failure, although likely less frequent than a link failure, can cause widespread disruption when it occurs. For instance, a node failure can completely isolate a region from communicating with others, as the flooding of several central offices caused by hurricane Floyd in 1999 [95]. More recently, several critical optical fibers were cut causing severe interruption of telecommunication services in all Eastern Asia due to Taiwan earthquake of December 26, 2006 [97].

According to [132], the fiber cut rate in Telcordia network is 4.39 cuts/year/1000 miles of fiber. Also, FCC published that 3 and 13 cuts for miles of fiber were experienced in metro and long haul networks respectively in 2002 [38]. These statistics show that the frequency of cable cut events can be up to thousands of times higher than node failures, therefore it is acknowledged that providing protection against these failure scenarios is a realistic goal.

Another common assumption is that of single link failures. Indeed, a comprehensive survey on cable cut events reported that the average cable repair time is 5.2 hours and all of 160 of the cable failures experienced were single-failure events [21]. This strengthens the claim that the probability that a second link failure arises during another failure repair is very unlikely.

Protection techniques involve providing some redundant capacity within the network and automatically rerouting traffic around the failure using this redundant capacity. We will focus on a recent development in transport network survivability: The *p-cycle* technique [39, 46]. p-Cycles are cyclic protection structures which offer very fast protection switching with guaranteed transmission integrity of protection paths. p-Cycle designs also achieve high spare capacity efficiency and support independent routing of traffic, without constraints implied by the placement of protection structures. Another operational advantage of p-cycles is that protection can be implemented on simple nodal equipment [65] and it does not require expensive optical switches. Among the interesting features of p-cycle configuration, we can cite:

– It can be logically managed on a per channel basis;
– It can be adapted to implement multi-priority protection;
– It can be adapted to changing traffic patterns;
– It can also be adapted for maximized survivability in multiple failure scenarios;
– It supports path length or optical reach restrictions.

Overall, p-cycles offer intriguing and promising alternatives to conventional optical network architectures. Hence, there is considerable motivation to further explore and refine this domain of networking technology and theory. Originally designed for link protection, several extensions of the p-cycle concept have been proposed by researchers. Among them, we remark the failure independent path protecting (FIPP) p-cycles which provide end-to-end path protection. Thus, FIPP p-cycles approach mesh-like efficiency in capacity utilization while retaining ring-like speed (see Chapter 2).

1.1 Motivation and objective

This thesis concerns the design of survivable optical networks using p-cycle schemes, more particularly, FIPP p-cycles, in link failure scenarios. Our study focuses on the placement of p-cycles assuming that the working routes for the set of connection requests are defined *a priori*. This problem is known as the non-joint optimization problem. In contrast, when the protection and working networks are determined simultaneously, we have the joint optimization problem. The main advantage of joint over non-joint optimization is that it leads to more resource-efficient designs. However, it corresponds to a much more complex design problem and the solution of medium size problem instances is not reachable. In addition, non-joint optimization is a realistic assumption since placement of protection capacity is considered as a strategic decision whereas working routing is an operational decision [108].

Since most existing work done on p-cycles concerns heuristic algorithms or methods suffering from critical lack of scalability (see Chapter 3), the objective of this thesis is twofold: on the one hand, to propose scalable solution methods enabling to approach larger problem instances, and on the other hand, to produce optimal or near optimal solutions with mathematically proven optimality gaps. For this, we rely on a powerful tool for solving large scale linear programming problems: Column generation [36, 37]. A column generation approach works with a reasonably small subset of the columns in the linear problem, originating the restricted master problem. This is solved iteratively, augmenting the number of columns until optimality of the original problem is proved with the available columns. Entering columns are found by solving an auxiliary problem, usually called pricing problem, which tries to identify variables with negative (or positive) reduced cost.

In this thesis, column generation is used as an intelligent way of enumerating promising cycles to be part of p-cycle designs. We also propose mathematical formulations for the master and pricing problems within the column generation algorithms. Good formulations are crucial to achieve an overall effective solution approach.

1.2 Thesis organization

The study carried out in this thesis originated several articles submitted to publication in scientific journals or international conferences with peer refereeing. Some of these articles have already been published or accepted for publication. This thesis consists of four main chapters, each one presenting an article selected among all those produced, and an additional chapter concerning a preliminary study on availability in p-cycle networks. The articles not included here present preliminary results, which were improved in the selected articles. Because these chapters must be presented exactly as the articles published or submitted to publication, there may be some redundancy throughout the thesis, which is organized as follows.

Chapter 2 provides background information on optical network survivability, which covers concepts and terms relevant to network survivability and introductions of various survivability techniques. The concepts and terms include network failure types, the difference between protection and restoration, etc. The survivability techniques are categorized into electronic and optical layer schemes and involve the popular point-to-point systems and self-healing rings as well as more advanced mesh survivable networks, which include classical link and path protection, and p-cycle schemes.

Chapter 3 presents a review of the existing solution methods found in the literature concerning p-cycle techniques, with a special emphasis on path-protecting p-cycles, which are the main topic of this thesis. Firstly, we divide the methods for link-protecting p-cycles into two categories according to the solution approach applied to optimize the design of p-cycle networks: those based on explicit enumeration or pre-selection of candidate cycles, and those applying alternative solution strategies. Then, the literature on FIPP p-cycle is covered.

In Chapter 4, we study the first column generation algorithm for the design of survivable networks based on FIPP p-cycles. Mathematical formulations for the master and pricing problems are proposed. The resulting algorithm obtains better solutions within reasonable running time in comparison with existing methods.

Chapter 5 is dedicated to a systematic comparison of p-cycles and classical shared protection schemes. Although these protection schemes have been compared in the literature, we perform an accurate comparison by using a unified column generation based approach to find provably good results.

Chapter 6 concerns an empirical evaluation of directed and undirected link- and path-protecting p-cycles under asymmetric traffic scenarios. We show how much additional protection cost results from employing bidirectional systems in such scenarios. Here again, column generation is used to obtain solutions with very small optimality gaps. The formulation presented in Chapter 5 is improved and a new formulation is proposed for directed FIPP p-cycles.

In Chapter 7, some different assumptions on the problem are considered and a much more compact formulation of the pricing problem is obtained. In addition, we also propose a new decomposition of the problem which greatly improves the efficiency of the whole algorithm and allows us to solve larger problem instances. As for integer solutions, two heuristic approaches are proposed.

A preliminary study on a column generation algorithm for the design of survivable networks using link-protecting p-cycles under availability constraints is presented in Chapter 8. Therein, we report the first lower bounds obtained for the problem and compare them with integer solutions existing in the literature.

Finally, Chapter 9 concludes the thesis and suggests future research directions.

1.3 Articles produced during the thesis

In the following, we present the chronological list of the articles published or submitted to publication in journals and international conferences with peer review during the thesis. Those included here as chapters are indicated by an star ($\star$).

1. $\star$ B. Jaumard, C. Rocha, D. Baloukov and W.D. Grover. A column generation approach for design of networks using path-protecting p-cycles. In *Proceedings of the International Workshop on the Design of Reliable Communication Networks (DRCN)*, October 2007.

2. C. Rocha and B. Jaumard. Revisiting p-cycles / FIPP p-cycles vs. shared link /

path Protection. In *Proceedings of the 17th International Conference on Computer Communications and Networks (ICCCN)*, August 2008.

3. C. Rocha, B. Jaumard, and P.-E. Bougué. Directed vs. undirected p-cycles and FIPP p-cycles. In *Proceedings of the International Network Optimization Conference (INOC)*, April 2009.

4. $\star$ C. Rocha and B. Jaumard. A column generation approach for shared protection schemes in WDM mesh networks. *Pesquisa Operacional*, 2009 (accepted for publication).

5. $\star$ C. Rocha, B. Jaumard, P.-E. Bougué. Asymmetry issues in p-cycle and FIPP p-cycle protection schemes, 2009 (submitted for publication in *Networks*).

6. $\star$ C. Rocha and B. Jaumard. A hierarchical decomposition method for efficient computation of path-protecting p-cycles, 2009 (to be submitted for publication in *Telecommunication Systems*).

7. M. Kiaei, A. Ranjbar, C. Rocha, B. Jaumard and C. Assi. Improved Availability Models for p-Cycle-Based Network Design. In *Proceedings of the International Workshop on the Design of Reliable Communication Networks (DRCN)*, October 2009 (to appear).

CHAPTER 2

FUNDAMENTALS OF NETWORK SURVIVABILITY

As previously discussed in Chapter 1, network failures such as fiber cuts and equipment fault can be catastrophic and survivability mechanisms are of paramount importance. This chapter provides a brief description of fundamental concepts in network survivability in the context of undirected networks (the directed case is fairly similar). First, we will introduce basic concepts and terms used in the domain. Then, a discussion of various survivability techniques, including p-cycle schemes, will be presented.

2.1 Basic concepts

2.1.1 Terminology

Some important terms in transport networks and survivability are summarized as follows. Although the terms found in the literature are not necessarily uniform, we follow what we consider to be the most common terminology.

Span: Physical entity that is a collection of all channels between two adjacent nodes. Typically, a span is a set of cables co-routed in the same ducts. Each cable may have multiple fibers, and each fiber may carry many multiplexed signals.

Channel: Wavelength on a fiber link.

Path: Route in the physical network.

Lightpath: All-optical path between a pair of nodes which may go through multiple fiber links, i.e., a path that optically bypasses intermediate nodes. Occasionally, we will also refer to it as a segment.

Request: Demand of traffic with a given bandwidth between two end nodes.

Connection: Capacity occupied by a request over a path.

Optical hop: Lightpath traversed by a connection request. The number of optical hops on a path can be computed from the number of signal regenerations across that path. For example, if a signal is submitted to two regenerations, then its path traverses three optical hops.

Working path: Path used in the “pre-failed“ state, i.e., carry traffic under normal operation conditions. Also called primary path.

Protection path: Alternate path to carry traffic in the ”failed“ state. Also called backup path.

Restoration time: Time elapsed between the moment at which a failure occurs and the moment at which traffic is restored.

Working capacity: Capacity used by the working paths on a link.

Protection capacity: Capacity used by the protection paths on a link. The term spare capacity is used to designate all capacity, other than working capacity, available on a link and it is more often used in the context of networks without capacity constraints.

Redundancy: Ratio of the total spare capacity to the total working capacity. It is often used to compare the performances of different survivability techniques.

Optical cross-connection: Switching action performed by a device called optical cross-connect (OXC) in order to setup a lightpath.

OC-n: Transmission rate (OC means Optical Carrier). OC-n corresponds to a $n\times$ 51,84 Mb/s signal, e.g., OC-48 $\simeq$ 2.5 Gb/s and OC-192 $\simeq$ 10 Gb/s.

Availability: Probability of finding the system in the operating state at any arbitrary time in the future.

2.1.2 Optical networks

Currently, there are basically two generations of optical networks. In the first generation, all the switching and other intelligent network functions were performed by electronics while optics were essentially used for network capacity and transmission. However, we are moving towards a second generation of optical networks, where some of the routing, switching, and intelligence is performed in the optical layer. Several advantages come from this change as it becomes more difficult for electronics to process the increasing amount of data. Moreover, the electronics at a node only need to handle the data addressed to that node while all the remaining data is routed through in the optical domain, significantly reducing the need for electronic equipment. These networks are based on WDM transmission and are called wavelength-routed networks, where WDM stands for wavelength-division multiplexing.

WDM networks are a type of high-speed transport networks, in which wavelength-division multiplexing is applied to simultaneously transmit multiple distinct wavelengths in a single fiber. Depending on the spacing between two neighboring wavelengths, we can have dense WDM (DWDM) or coarse WDM (CWDM). Recently, the International Telecommunication Union (ITU) has standardized a 20nm channel spacing grid for use with CWDM, while DWDM systems use 100 GHz (0.8nm), 50 GHz (0.4nm) or even 25 GHz channel spacing [55]. The new ITU specification opens for 18 CWDM channels on a special type of fiber [56], while a grid of 81 wavelengths is defined for DWDM systems.

WDM systems use different wavelengths for different channels. Each channel may transport homogeneous or heterogeneous traffic, such as SONET/SDH (synchronous optical network/synchronous digital hierarchy) over one wavelength, ATM (Asynchronous Transfer Mode) over another, and yet another may be used for TDM voice, video or IP (Internet Protocol). WDMs also makes it possible to transfer data at different bit rates. Thus, it offers the advent that one channel may carry traffic at OC-48, OC-192 or up to OC-768 rate while another channel may carry traffic at a different rate transmission; all on the same fiber. The technology applied to a WDM network node must support some functionalities, among which wavelength routing (or switching) and multiplexing/demultiplexing are the most important ones.

The optical layer provides lightpaths for use by its client layers, such as SONET, IP, or ATM layers. Lightpaths are all-optical connections from a source node to a destination node over a wavelength, with optical bypasses on each intermediate link. At intermediate nodes, the lightpaths are routed and switched from one link to another and, in some cases, lightpaths may be converted from one wavelength to another as well along their route if we have all-optical wavelength conversion capabilities within the network. Different lightpaths in a wavelength routing network can use the same wavelength as long as they do not share any common links.

2.1.3 Protection versus restoration

A survivable network can basically use a protection or a restoration scheme. Protection refers to a preplanned system where a protection path is precomputed for each potential failure during network design or at the time of connection establishment. In the event of a failure, the disrupted connections are recovered by using the reserved network resources for failure recovery, which can be dedicated for specific failure scenarios or shared among different ones. In contrast, restoration schemes take action in real time which means that the backup route is computed after the failure occurrence using the available network resources, based on the failure and the state of the network.

Generally, restoration schemes are more efficient in utilizing network capacity because no resources are allocated before any failure occurs. They do not allocate spare capacity in advance, but protection schemes have faster restoration time as there is no dynamic search for spare network resources and they can always guarantee recovery from failure, which is not the case in restoration as the network may not have sufficient spare resources at the time of a failure event.

2.1.4 Dedicated versus shared protection

Protection schemes can be further classified into dedicated or shared protection. In dedicated protection, each working connection is assigned its own dedicated spare capacity in the network over which it can be rerouted in case of a failure. A typical example of dedicated protection is 1+1 protection, in which the optical signals are transmitted simultaneously on two dedicated channels between end nodes. In shared protection, the reserved network resources can be shared among multiple working connections under the assumption that they do not fail simultaneously.

It is clear that dedicated protection is very fast in service recovery since both working and protection paths are fully set up in advance. Besides, it can handle multiple failures as there is no resource sharing among connections. However, shared protection is more efficient in resource utilization because it reduces the amount of bandwidth needed in the network for protection. Another advantage of shared protection is that the protection bandwidth can carry low-priority traffic under normal conditions. When the bandwidth is needed to protect a connection in the event of a failure, this low-priority traffic is preempted for restoration of a higher priority request.

2.2 Survivability techniques

Survivability schemes are extensively used to maintain or restore an acceptable level of performance during network failures by applying various restoration techniques. For instance, in WDM networks, the failure of a network element, such as fiber link, cross-connect, etc., can cause the failure of several lightpaths, thereby leading to large data and revenue losses [38]. Three important criteria are usually considered when evaluating a survivability mechanism: capacity efficiency, operational complexity, and restoration time. These aspects may greatly vary depending on the network layer in which survivability is applied.

Survivability can be provided within different network layers, such as IP, ATM, SONET, and the optical layer. Despite the existence of widespread survivability mechanisms in the client layers, it is very attractive to provide fault recovery in the optical layer according to the following main reasons [81, 95]:

– The optical layer can provide survivability functions that the higher layers, such as IP and ATM, do not provide.
– Optical layer protection can yield significant cost savings in comparison to client layer protection, as shown in [95].
– An additional level of resilience can be provided with optical layer survivability (for example, against multiple failures).
– Some faults can be handled more efficiently in the optical layer than in the client

layers.

– In general, it is preferable to recover a failure in the layer in which it occurs.

– Possibility of reducing the restoration times below those obtained with SONET/SDH [84].

Despite the mentioned advantages, optical layer protection also have its own limitations. For example, client equipment faults need to be solved by the client layer and traffic at a granularity finer than one lightpath cannot yet be handled in the optical layer.

In this thesis, we will focus on optical layer techniques and only briefly discuss techniques within the electronic layer, such as SONET/SDH. A more detailed review of survivability techniques for optical networks can be found in [38, 95, 120, 136].

2.2.1 Electronic layer schemes

In this section, we briefly discuss the two most commonly used survivability techniques in the electronic layer: Point-to-Point Systems and Self-Healing Rings. These techniques are inherently protection schemes featuring the following relevant characteristics:

(i) Prior to any failure, the protection paths are pre-determined and pre-configured.

(ii) Moreover, only the nodes adjacent to the failure performing switching action, which offers the great advantages of fast restoration and simple operations.

However, these techniques suffer from low spare capacity efficiency. Indeed, they require over 100% spare capacity redundancy. A more detailed review of these techniques can be found in [136], [95], and [38].

2.2.1.1 Point-to-point systems

Point-to-point systems use automatic protection switching (APS) to switch failed traffic to protection facilities. It consists of two fundamental types of protection mechanisms: 1+1 and 1:1 protection. The former is the simplest and fastest protection technique because there is no signaling between nodes. The traffic is transmitted simultaneously on both working and dedicated backup fibers (or channels) and the receiver selects the signal with better quality. In 1:1 protection, there are also two channels from source to destination but the traffic is only transmitted on the working fiber. Upon a failure event, the backup fiber is used to carry the affected traffic. Note that the presence of co-routed working and protection fibers in these schemes is unsuitable for protection against cable cuts. For this reason, diverse routing has been imposed to APS systems in order to obtain path diversity between working and backup fibers. Please refer to [4] for an extensive discussion on routing and diversity algorithms for such systems.

As signaling is required between the source and destination, 1:1 protection yields a longer restoration time in comparison to 1+1 protection. However, it offers two main advantages over 1+1 protection. First, the backup fiber can be used to transmit lower priority traffic under normal operation. Another advantage is that 1:1 protection can be extended to 1:N (or M:N) protection in order to achieve better capacity efficiency, by allowing multiple working fibers to share a single (or multiple) common backup fiber. Even in shared versions of point-to-point APS, the backup fibers are dedicated to the working fibers they protect, which is becoming prohibitively expensive due to the frequent changes and augmentation of Internet traffic.

2.2.1.2 Self-healing rings

Self-Healing Rings are very popular carrier structures used in SONET/SDH networks since they incorporate protection mechanisms which automatically detect failures and switch the failed traffic. Unidirectional Path-Switched Ring (UPSR) and Bidirectional Line-Switched Ring (BLSR) [38, 136] are the two most popular ring-based techniques.

Under UPSRs, a ring carries working traffic in only one direction while another ring is dedicated to protection, as shown in Figure 2.1. Indeed, the traffic is sent simultaneously on both working and protection fibers but in opposite directions. As in 1+1 APS systems, the destination node selects the best signal.

Instead of switching over end nodes as in UPSRs, BLSRs perform local switching at the nodes adjacent to the failure. They can be of two types: a 4-fiber or 2-fiber structure. In a BLSR/4, two fibers carry working traffic in both directions along the ring

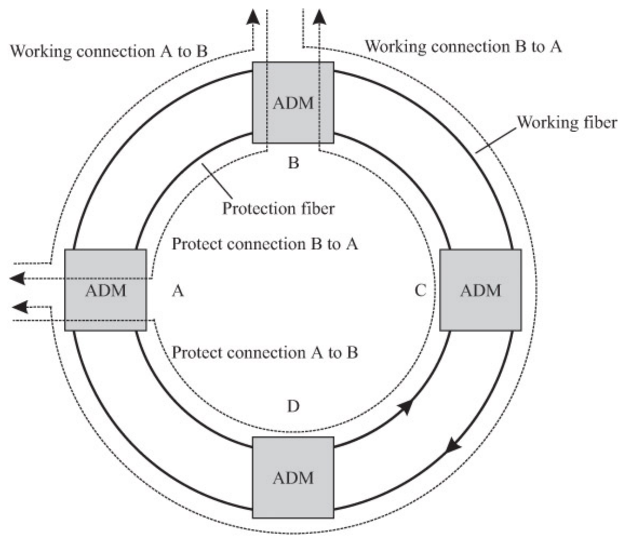


Figure 2.1: A unidirectional path-switched ring. Source: [95].

and another two fibers are used for protection, as shown in Figure 2.2. Upon a failure, the traffic is partially carried through the working fiber and the ring performs a loopback function [29] to switch the traffic to the protection fiber around the failed link. In a BLSR/2, half of the capacity of each fiber is used by the working traffic and the other half is dedicated to protection.

Ring networks are usually simple to manage and have very short restoration time because they rely on pre-configured protection structures, which is of great interest in many applications. Nevertheless, they yield excessive redundancy of spare capacity and are expensive to upgrade. Another drawback is that both working and protection traffic are constrained to be over a ring structure which may be prohibitive in large scale mesh networks. A detailed discussion of ring-based protection schemes can be found in [95, 136], and a recent survey of mathematical programming models for ring-network designs can be obtained in [13, 121].

2.2.2 Optical layer schemes

The WDM-based equivalent architectures to SONET/SDH rings are Optical Path Protection Rings (OPPR) [86] and Optical Shared Protection Rings (OSPR) [78]. OPPR

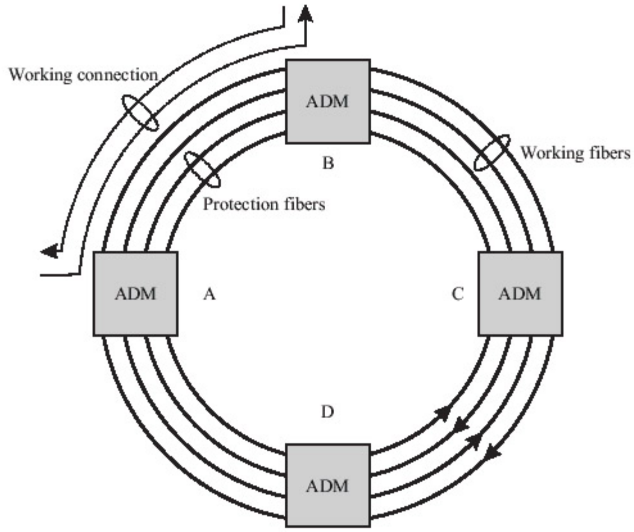


Figure 2.2: A four-fiber bidirectional line-switched ring. Source: [95].

is similar to UPSR, except that it operates at the optical layer, and OSPR is analogous to BLSR/4 with some changes.

Given the well-known disadvantages of ring networks, we will turn our attention to survivability mechanisms for optical mesh networks, whose classification is summarized in Figure 2.3. Mesh survivable networks refer to networks over a mesh-like physical topology where survivability techniques can find arbitrary backup paths. Depending on their characteristics and structures, these survivability schemes can provide protection against node and/or link failures. Furthermore, they can be divided into link-based, path-based, and segment-based techniques.

Link-based schemes

Originally devised for ring networks, link-oriented protection has migrated to mesh networks. Link-based techniques act locally in the vicinity of the failed link to switch the traffic to protection paths between the end nodes of the failed link. Therefore, only these nodes need to be notified of the failure.

These schemes can be implemented in two ways: Link protection and link restoration. In link protection, for each link of a working path, a protection route and the assigned capacity are pre-determined and stored at the end nodes of the link. Because multiple backup paths may share the spare capacity reserved on each link, the backup paths are set up only after the failure occurs so that restoration is achieved. If a dedicated strategy is used instead, all the backup paths are ready to be used at any time. In link restoration, not only the backup paths are configured and set up, but also all the protection routes and spare capacity are determined and allocated by the network in real time whenever a failure occurs.

Figure 2.4(a) illustrates how protection is provided within this scheme. For example, if a failure on link B-C occurs, the traffic going through this link is rerouted over a backup route and then continue its way over the subsequent working links. Each working link (solid lines) has its own protection path (dashed lines). From this example, we can also see that it corresponds to a shared protection scheme as links C-D and D-E have a link in common in their protection routes.

These schemes provide very fast restoration due to fault localization but, on the down side, they are less efficient in utilizing network capacity in comparison to path-oriented approaches because they act locally. Link-based survivability mechanisms include the classical Shared Link Protection (SLP), Dedicated Link Protection (DLP), and Link Restoration (LR) as well as link-protecting p-cycles, which are discussed in Section 2.3.

Path-based schemes

Path-based schemes consist in protecting each connection individually by providing an end-to-end backup path. The techniques in this category include Path Restoration (PR) [54], Shared Path Protection (SPP), also known as Shared Backup Path Protection (SBPP) [62], and path-protecting p-cycles (see Section 2.3). Such techniques differ in the way backup paths can be routed, as discussed below.

When a link fails, PR does not impose any restriction regarding the backup path for a given working path, except that it cannot use the failed link. This means that different protection paths can be used to restore the traffic through a given working path (e.g., a different protection path for each traversed link) and they are not restricted to be link or node-disjoint from the working path as in Figure 2.4(c). In this restoration scheme, the

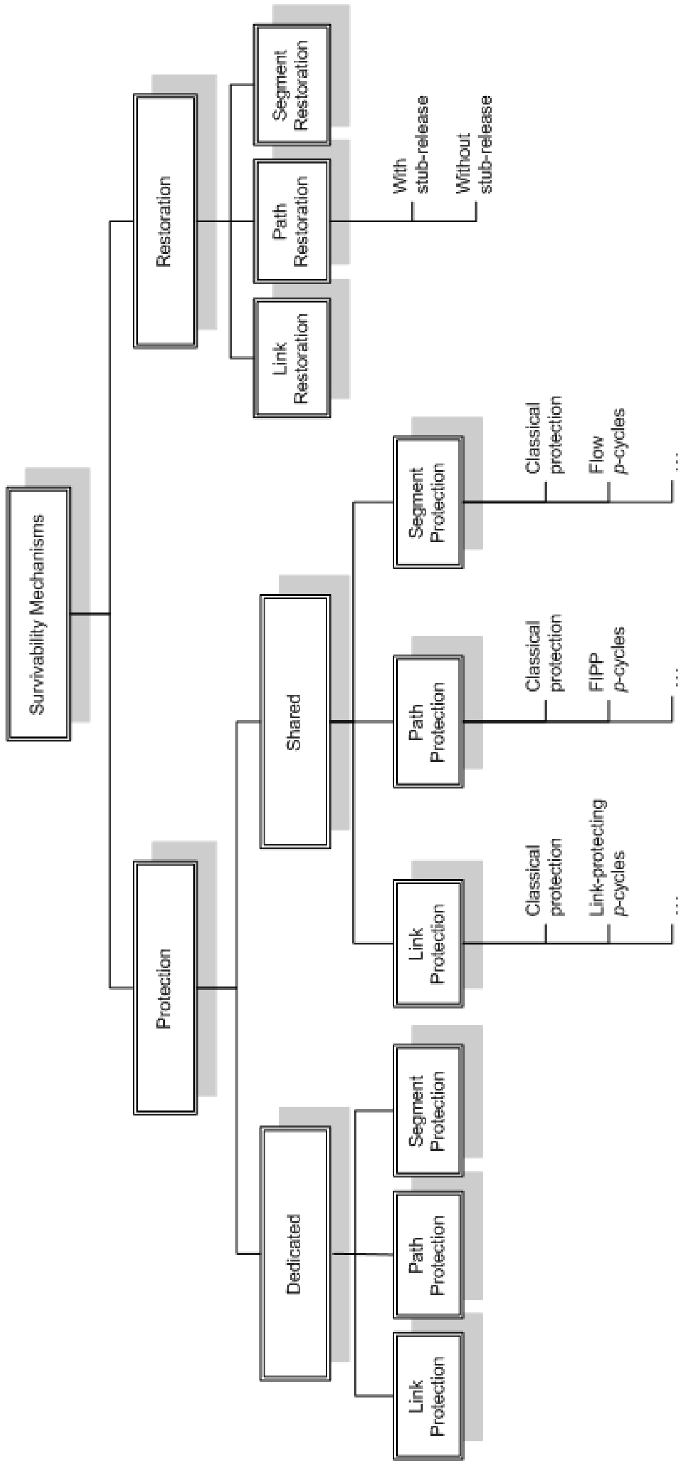
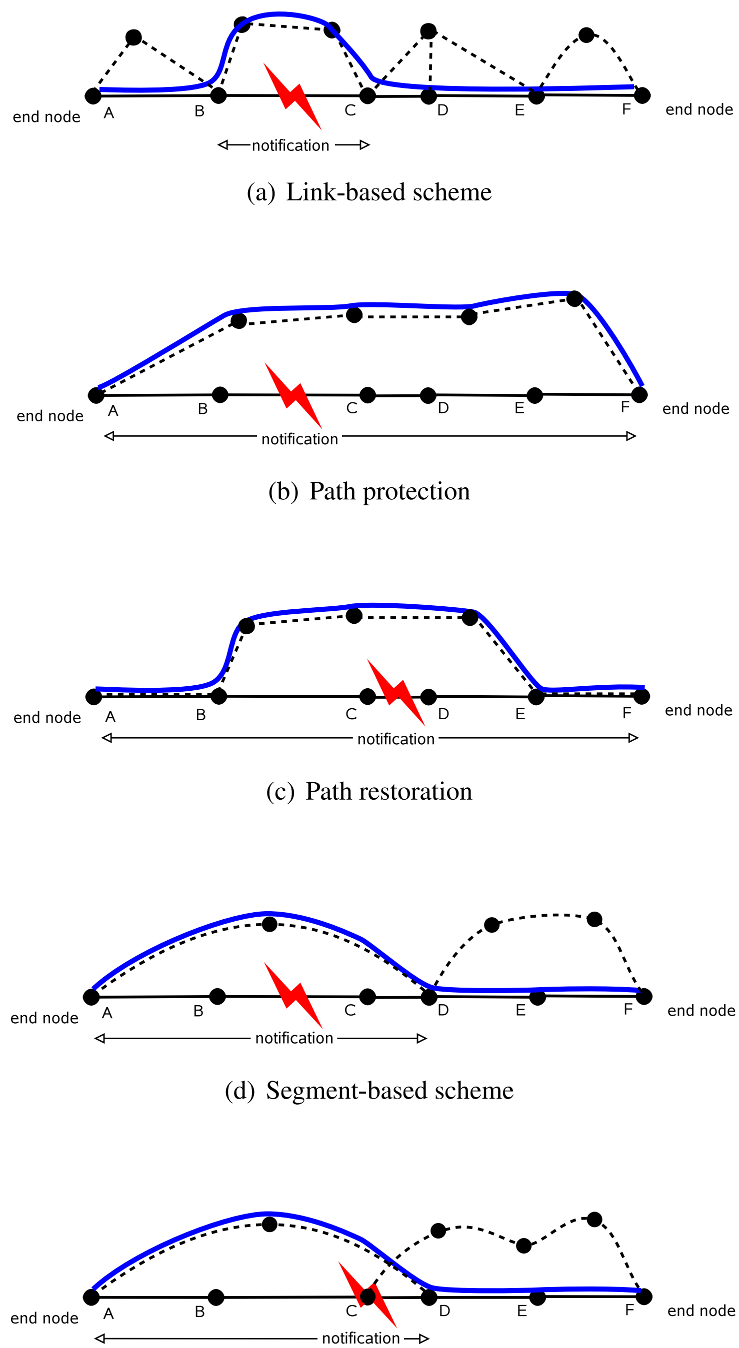


Figure 2.3: Classification of survivability mechanisms for optical mesh networks.



(a) Link-based scheme

(b) Path protection

(c) Path restoration

(d) Segment-based scheme

(e) Overlapping-segment-based scheme

Figure 2.4: Mesh survivability mechanisms. Working and protection links are represented by solid and dashed lines respectively. Thick solid lines represent the route taken by the affected traffic after restoration.

end nodes of each connection affected by a failure dynamically determine the protection route and spare capacity needed for restoration. PR can be further divided in two sub-cases: Restoration with and without stub release [54]. Here, “stub” is referred to as the surviving parts of an affected working path. For example, in Figure 2.4(c), parts A-B-C and D-E-F are the stubs of the defective working path. Path restoration without stub release does not release the capacity on the unharmed parts while, in the stub-release version, the capacity on the two stubs are released and can be used for protection. PR with stub-release is assumed to achieve the best spare capacity redundancy among all survivability techniques at the price of a more complex protection scheme with longer restoration time.

SPP allows only one protection path to restore the traffic on a given working path, regardless of the failure location. Besides, SPP requires the backup path to be disjointly routed from its working path, as shown in Figure 2.4(b). The backup path can be link or node-disjoint from its corresponding working path depending on the type of protection to be provided. In case of a link failure, a notification signal is sent to the end nodes of each connection traversing the failed link in order for them to switch the traffic over the working path to the backup path. Although less capacity efficient than path restoration, SPP yields a faster restoration time because it is a failure-independent survivability scheme.

The recently proposed Demand-wise Shared Protection (DSP) mechanism [68] combines advantages of both dedicated and shared path protection. DSP is based on diversification routing scheme [22] which consists in routing the traffic volume of a demand (request) though several different working paths. In DSP, the spare capacity is shared by the working paths of a same demand, but not among different demands.

Segment-based schemes

Segment-based schemes are based on a concept generalized from the two previous schemes. These schemes consist in dividing each working path into a sequence of path segments, which can overlap or not, and protecting them separately. As it is a generalized concept, segment can be used to model a complete path itself or a single link. When a failure occurs, only the affected segment performs protection switching and the other unaffected segments are oblivious to the failure. In the classical segment protection, the working segments are concatenated but not overlapping, as illustrated in Figure 2.4(d). As in link-based schemes, segment-based schemes are not able to protect end nodes of the segments. However, they have the advantages of faster restoration and more spare capacity efficiency compared to path-based schemes, despite the complexity in network planning and operation.

Protection using overlapping working segments was introduced in [93] and further developed in [51, 52, 129]. An enhanced variant with overlapping protection segments was introduced in [7]. The most important advantage of this scheme over the classical segment protection is that it provides recovery against node failure, as shown in Figure 2.4(e), although they consume more spare capacity.

Path-Segment Restoration (PSR) [50] [116], Shared Segment Protection (SSP) [116], Short-Leap Shared Protection (SLSP) [52] and flow p-cycles (see Section 2.3) are all segment-based schemes.

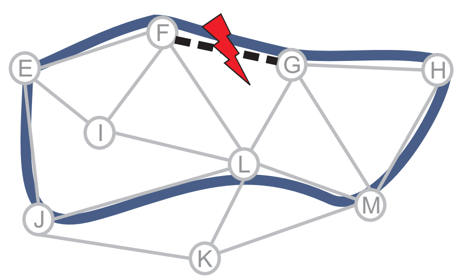
2.3 p-Cycle techniques

2.3.1 Link-protecting p-cycles

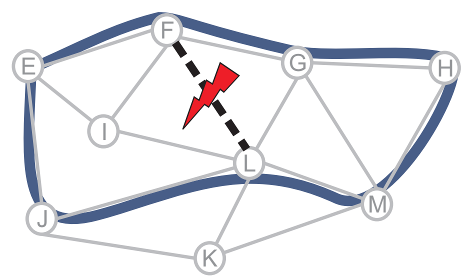
p-Cycles are fully preconnected cyclic protection structures with preplanned spare capacity. They were introduced in 1998 by Grover and Stamatelakis [39]. When a link fails, only the two end-nodes of the link perform protection switching, therefore no switching actions are required at any intermediate node of the cycle. Unlike rings, p-cycles protect against straddling link (chord) failures as well as failures on links over the ring itself. Besides, under p-cycles, the working paths are routed independently, i.e., they are not restricted to follow a cyclic structure. These characteristics make p-cycle based networks much more capacity efficient than ring-based networks, while providing "ring-like" speed switching [46].

Figure 2.5 illustrates the operation of basic link-protecting p-cycles. A same single p-cycle, with one channel of spare capacity, is shown by the thick solid line in both

Figure 2.5(a) and (b). In Figure 2.5(a), a link on the cycle fails (dotted line) and the surviving part of the cycle is used to provide a protection path (arrowed solid line), just like rings. In Figure 2.5(b), a straddling link is protected by the same p-cycle. Each unit of spare capacity on a p-cycle can protect two units of working capacity on a failed straddling link because two protection paths are provided in this case. In the example, halves F-E-J-L and F-G-H-M can be used to protect two units of working traffic on link F-L.



(a) On-cycle link failure



(b) Straddling link failure

Figure 2.5: A link-protecting p-cycle.

So far, basic link-protecting p-cycles have been described. However, the p-cycle concept has been extended to more elaborate techniques, such as segment-protecting and path-protecting p-cycles, which are presented in the next two sections. Other extensions have recently been proposed [38, 40] but will not be discussed in this thesis.

2.3.2 Flow p-cycles

Segment-protecting p-cycles, shortly called flow p-cycles, were introduced by Grover and Shen in [45, 118]. They are an extension of the p-cycle concept to cover straddling flows. A flow (or segment) is any single contiguous segment of a working path. The concept of flow p-cycles is illustrated in Figure 2.6, where we consider three working paths (4-3-5-9, 4-2-7-6-9, and 1-0-7-6-10) and one cycle. In case of failure of links 2-7 and 6-7, the traffic cannot be protected by the cycle if it is a link-protecting p-cycle. But, under a flow p-cycle, the contiguous flow or segment 2-7-6 can be protected by two alternate routes, 6-8-0-2 and 6-5-3-2. Moreover, in case of failure of intermediate node

7, the flows between node pairs 1-10 and 4-9 can be protected by the flow p-cycle, but the flows must be unchanged in its composition between the nodes where it intersects the cycle.

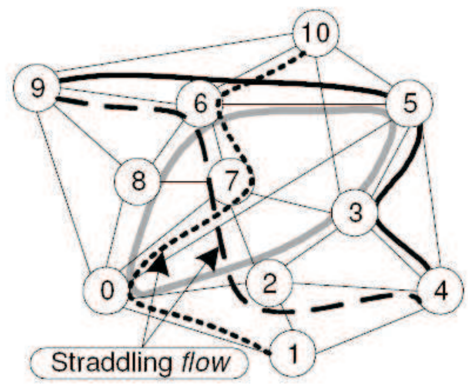


Figure 2.6: Flow-protecting p-cycle. Source: [118].

There are several types of intersections between a working path and a candidate flow p-cycle, possibly involving more than two intersection nodes. These intersections may be arbitrarily complex, possibly leading to a complex flow p-cycle design. Depending on each possible span failure, each p-cycle must know which failed segments are under its protection. This is achieved by maintaining preplanned information at each node for each flow p-cycle traversing it. In other words, failure detection is required at each node for proper activation of the flow-protecting p-cycles.

2.3.3 FIPP p-cycles

Basic link-protecting p-cycles were further extended with the goal of providing end-to-end path protection without requiring any failure specificity issues, originating the Failure Independent Path-Protecting (FIPP) p-Cycles. Under FIPP p-cycles, the cyclical protection structures can be shared by a set of working paths for protection as long as the working paths in this set are mutually disjoint or, if they are not, their protection paths must be mutually disjoint. If these criteria are met, there will be no contention for spare capacity after a failure. Furthermore, the end nodes of the working paths must also be crossed by the cycle assigned to protect them. According to the properties of p-cycles, a straddling working route may have up to two protection routes, because no link on the cycle is affected in case of a failure.

Without loss of generality, the concept of FIPP p-cycles will be explained by imposing mutually disjoint routes in the set of paths protected by the same FIPP p-cycle. Let us consider the example illustrated in Figure 2.7. This example shows how this set of mutually disjoint paths can share the spare capacity of a single FIPP p-cycle, without addressing failure location. Only one route will require protection under a single failure scenario because of the disjointness property of the set. Some of the routes in the example fully straddle the FIPP p-cycle, such as B-D-H, E-I-L and A-F-G-H. These routes can have two working paths protected per unit of spare capacity on the cycle. In addition, there are some routes lying fully over the cycle (A-B-C and L-M-H) and others partially over the cycle (E-J-K-M and C-H-M-N).

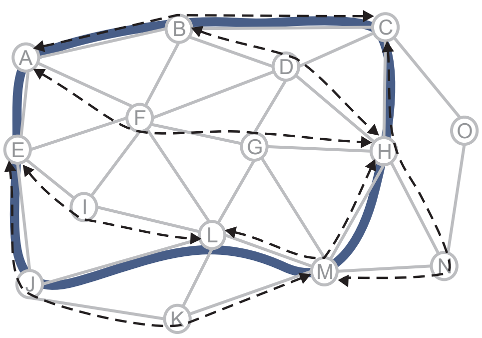


Figure 2.7: A FIPP p-cycle shared by a set of pairwise disjoint working paths.

FIPP p-cycles and SPP are similar in the sense that working routes allowed to share spare capacity on backup routes must be pairwise disjoint, however, there are significant differences between them. First, the routes protected by the same p-cycle share a fully pre-connected protection structure instead of protection channels. Also, SPP does not allow partial or full overlap of a working path with its own backup path, like with partially or fully on cycle routes in FIPP p-cycles as shown in Figure 2.7.

The main properties of FIPP p-cycles, presented in [64], are enumerated as follows:

1) Only cross-connections at the end nodes are needed in real time to compose the

protection paths which result in fast restoration.

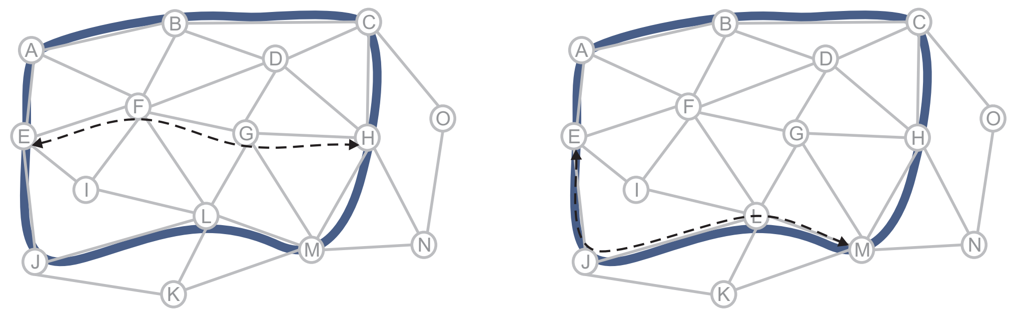
2) The protection paths are fully pre-cross-connected, providing certainty about functioning in case of a failure.
3) Protection switching is end-node controlled, entirely failure-independent, and can recover either link or node failure along the path. Only a single switching action is pre-programmed at each end-node.
4) Straddling routes can have two working paths protected by each unit-capacity p-cycle.
5) Node-failure protection is achievable if working routes are node (and consequently link) disjoint. Node-disjointness can be relaxed to link-disjointness if only link failure is required.
6) Protection paths are known in advance before failure, thus, their length can be easily limited by restricting the size of eligible cycles.

The logical operations of FIPP p-cycles are explained as follows by considering the different protection relationships of a given FIPP p-cycle to working paths.

Straddling routes have no link in common with the p-cycle, as shown in Figure 2.8(a), where the working route and the cycle is represented by dashed and solid connected lines, respectively. In this case, two distinct protection paths are available on the cycle, and thus up to two working paths on this route can be protected in case of a failure. In case of a failure, only the end-nodes of the route perform switching actions and any criterion can be adopted to assign working paths to unique protection paths. The pre-assigned direction is stored at the end-nodes where the switching action takes place as soon as a working path failure is detected. Note that the predefined switching action does not depend on the type or the location of the failure.

The pure on-cycle relationship arises when all links of the working route are crossed by the cycle, as shown in Figure 2.8(b). This is the direct extension of the on-cycle concept from link-protecting p-cycles. The protection path for such a relationship is unambiguously determined as the complementary part defined by the links of the cycle that are not shared by the working route.

The working routes may also be partially over the cycle, which arises from the ex-



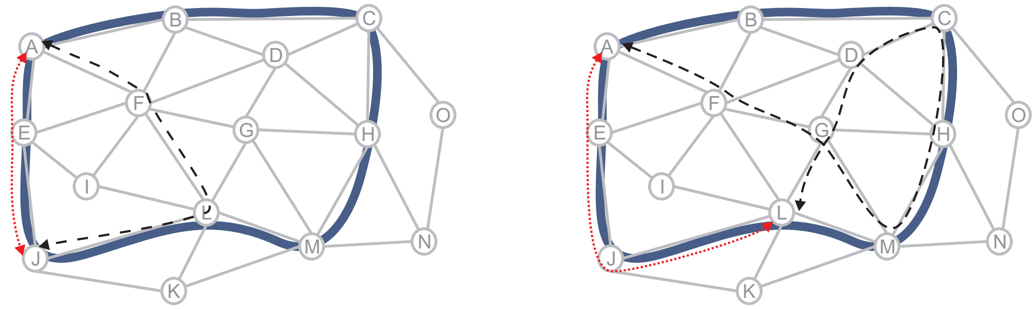
(a) Straddling working route

(b) On-cycle working route

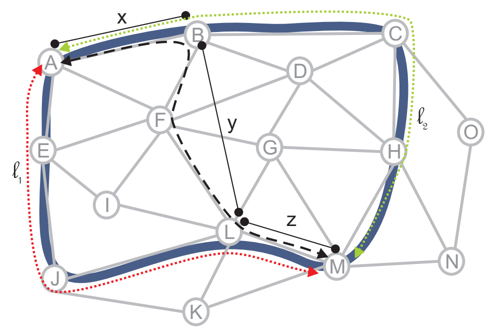
Figure 2.8: Different relationships between working routes and FIPP p-cycles.

tension to path protection and is not faced by basic p-cycles. This relationship occurs when at least one but not all links in the working route are shared by the cycle. There are two different types of this relationship according to its required operation. In the first one, the working path and the protection path provided by the cycle are disjoint. Two examples are illustrated in Figures 2.9(a), where the dotted arrowed lines represent the available protection path. Operationally, this is the same as the pure on-cycle relationship where the assigned protection path is enabled regardless of where the failure affects the working path.

The second type of partially on-cycle relationship occurs when, for a given working route, we cannot find a protection path on the p-cycle such that it does not share at least one span with its respective working route, as illustrated in Figure 2.9(b). In this case, two protection paths must be considered for protection and the switching logic is performed as follows. In the example, let us assume that path ℓ_1 (A-E-J-L-M) between nodes A and M is set as the default pre-assigned protection path for working path A-B-F-L-M. If segments x (A-B) or y (B-F-L) are affected by a failure, protection path ℓ_1 will survive and the behavior is the same as before. However, a failure on segment z (L-M) implies that the protection path defined by the default direction also fails and the affected working path must now be protected by path ℓ_2 (A-B-C-H-M). Fortunately, this can be realized locally at the end nodes simply by determining which side of the cycle was affected by the failure along with the working path. Thus, in case of coincidence



(a) Disjoint working and protection paths.



(b) Non-disjoint working and protection paths

Figure 2.9: Partially on-cycle working routes.

of failure states on both working path and its pre-assigned protecting path, the surviving protection path is selected for recovery.

CHAPTER 3

LITERATURE REVIEW

This chapter presents a literature review on p-cycle techniques, more particularly on path-protecting p-cycles which are the main topic of this thesis. Section 3.1 overviews basic link-protecting p-cycles, while Section 3.2 covers the current literature on FIPP p-cycles. In Section 3.3, we review the literature on the design of p-cycle networks with availability requirements. Finally, Section 3.4 summarizes some further works dealing with different design issues on p-cycles.

For a survey on the p-cycle concept and its extensions as well as existing solution methods for related problems, we refer the reader to [38, 47]. Additionally, various references in the literature [38, 81, 95, 119] can provide a general overview and discussion on the survivability mechanisms discussed in the previous chapter.

3.1 Link-protecting p-cycles

This section broadly divides the works on link-protecting p-cycles into two subsections according to the solution approach applied to optimize the design of p-cycle networks. Most of these works propose to enumerate or pre-select candidate cycles before applying integer linear programming (ILP) or any other solution approach. Firstly, a review of these works is presented, and then we discuss the existing alternative solution strategies.

3.1.1 Works based on cycle enumeration

The first paper on p-cycles was published by Grover and Stamatelakis in 1998 [39]. The authors present an ILP formulation for the design of fully restorable networks based on p-cycles. First of all, the set of connection demands are routed using the shortest path or any other routing scheme and the set of all simple distinct cycles up to some limiting size is generated from the network topology. Then an ILP model for minimizing the total protection capacity is solved. Designs obtained for five networks show that no or little additional capacity is needed in relation to a network with classical link protection. The main drawback of this approach is that it requires enumeration of all possible p-cycles. Because the number of cycles grows exponentially with the network size, only networks of moderate size can be solved to optimality.

To overcome this problem, researchers have proposed different strategies for pre-selecting a reduced number of "promising" p-cycles, thereby reducing the number of candidate cycles in the formulation [31, 42, 71, 118, 138]. Grover and Doucette [42] compare results of non-joint optimization and joint optimization, though the model for joint optimization is not provided. Two pre-selection metrics based on insights about efficient cycles are used: *topological score* (TS) and *a priori efficiency* (AE). The TS of a given cycle p is given by the total amount of protection provided by the cycle, i.e., the sum of the number of protection paths that the cycle can provide for each link in the network:

$$TS(p) = \sum_{\ell \in L} x_{p\ell},$$

where L denotes the link set, and $x_{p\ell}$ is the number of protection paths that cycle p can provide for link ℓ ($x_{p\ell} = 1$ if ℓ is an on-cycle link, and $x_{p\ell} = 2$ if ℓ is a straddling link). The AE metric is the ratio between the TS and the cost of the cycle:

$$AE(p) = \frac{\sum_{\ell \in L} x_{p\ell}}{\sum_{\ell \in p} c_\ell},$$

where c_ℓ is the cost of a unit of capacity on link ℓ. These metrics select cycles based purely on the topology of the network and do not take into consideration the traffic distribution. As a result, cycles offering little or no protection to the aimed traffic may be selected while cycles offering more significant protection may not. This shortcoming is alleviated in [118] where a new selection strategy based on the amount of protection offered by the cycles to the applied traffic is presented. Although both methods reduce the number of candidate p-cycles, they still require enumeration of all cycles in the network. Moreover, there is no optimality guarantee and no valid LP bounds to evaluate the quality of the obtained solutions.

Zhang and Yang [138] propose a simple algorithm, which they call Straddling Link Algorithm (SLA), for enumerating a small subset of cycles in the network. Here, no efficiency metric is used. Instead, the algorithm is based on the idea that one can see a cycle as a combination of two node-disjoint paths between the end nodes of a straddling link. Thus, they use Dijkstra's algorithm for finding disjoint shortest paths between end nodes of potential straddling links in order to compose candidate p-cycles. However, the cycles generated by SLA are generally inefficient because they usually have only one straddling span.

In [26], Doucette *et al.* propose a heuristic algorithm for the non-joint p-cycle optimization problem. The algorithm, called Capacitated Iterative Design Algorithm (CIDA), first compute a set of candidate p-cycles and then iteratively chooses one p-cycle from the set to reduce the current unprotected working capacities until all working capacities are protected. The selection of p-cycles is based on a new metric which depends not only on the number of on-cycle and straddling links, but also on the unprotected working capacity on these links, unlike a priori efficiency. This metric is called actual efficiency and is defined as

$$E_w(p) = \frac{\sum_{\ell \in L} w_\ell x_{p\ell}}{\sum_{\ell \in p} c_\ell},$$

where w_ℓ is the current unprotected working capacity on link ℓ. At each iteration, the actual efficiency of unused cycles needs to be recalculated as the unprotected working capacities change. The set of candidate p-cycles is generated by using SLA [138]. Then various operations (such as Expand and Grow) are performed in order to improve the initial set of cycles. The authors show that the resulting set of p-cycles have higher average a priori efficiency than those generated by SLA. Consequently, a better performance is obtained when this set of p-cycles is used by CIDA or an ILP method, at the cost of a larger amount of candidate cycles. Moreover, they compare the solution quality of their algorithm to designs obtained with a "pure" ILP method which takes the set of eligible cycles as input, obtaining gaps against optimality that raise up to 20%.

Another algorithm to compute an initial set of candidate p-cycles is proposed in [71].

The idea consists in generating a combination of high efficiency cycles and short cycles so that both densely distributed and sparsely distributed working capacities can be efficiently protected by the candidate cycles. Unlike the previous works, the number of generated p-cycles is not fixed but controlled by the algorithm by adjusting an input parameter. The first step of the algorithm is based on Johnson's cycle enumeration algorithm [59], which is based on depth first search in the graph. Weights are assigned to the links in order to guide the search. Then, the algorithm computes two short cycles for each link in the network by using a shortest path algorithm, one containing the link as an on-cycle link and the other with the link as a straddling link. The set of candidate cycles are used as input to CIDA algorithm [26] and the ILP formulation given in [38, 39]. Compared to optimal solutions, the obtained results show optimality gaps as large as 21.9%.

In a recent work by Eshoul and Mouftah [31], a new metric for preselecting high merit cycles, called Route Sensitive Efficiency (RSE), is introduced. The metric ranks each cycle according to the number of links of the working paths it can protect. The paper also proposes methods to approach the joint and non-joint optimization of the survivable routing and wavelength assignment (RWA) problem under static traffic using link-protecting p-cycles. The RWA problem, also know as lightpath establishment, consists in selecting a route for each demand connection and assigning a suitable wavelength. In the non-joint approach, the minimum backup capacity for placing p-cycles to protect against any single link failure is set up first, regardless of the amount and the distribution of the traffic. Then the RWA problem is solved by considering a subset of candidate routes for each source and destination pair and using an ILP formulation. In the joint version, the problem is formulated as in [38, 127] for both assumptions of full wavelength conversion and no wavelength conversion. Here again, a subset of candidate routes as well as candidate cycles are provided as input for the ILP model.

Other authors have also dealt with the wavelength conversion issue [82, 83, 104, 110]. Schupke *et al.* [104] developed ILP models for p-cycle configurations in WDM networks with and without wavelength conversion and applied then to a pan-European network. The non-joint optimization problem is considered and the authors use two cost metrics for determining the shortest paths for the demands. The set of candidate p-cycles is generated from the spare capacity using a breadth first search algorithm starting from each link. Moreover, the cycles are length-restricted and unidirectional, which means that straddling links can be protected in only one p-cycle direction. However, the authors avoid generating all possible cycles since the set of candidate cycles still increases exponentially with the network size and traffic distribution. In [110], the authors continue the work done in [104] and investigate the configuration of p-cycles in WDM networks with limited wavelength conversion. A relevant finding is that the total number of converters required for the network as a whole can be greatly reduced by only a small increase in protection capacity.

Mauz [82] revisits the joint p-cycle design problem for WDM networks with full wavelength conversion using an ILP, and proposed a heuristic for the problem with no wavelength conversion. As in [104], the designs are based on unidirectional p-cycles. The proposed heuristic uses the solution found for the problem with full wavelength conversion to derive a solution for the case with no conversion. The algorithm is based on the first-fit method to assign wavelengths [48] and the TS metric to choose the cycles to be included in the design. As a result, the author shows that networks without conversion require 40-60% more capacity than networks with full conversion. In [83], Mauz investigates the capacity of p-cycles using an ILP formulation for the joint design problem. A case study results in redundancy values of 60-130% for three European networks with degree 4.3-2.4 [80].

Besides the mentioned works proposing heuristic algorithms, we can also select from the literature the works of [73, 89, 123, 142]. In the technical report [123], Stamatelakis and Grover indicate that p-cycles are efficient from the structural point-of-view. They use a public-domain genetic algorithm package [19] to find arbitrary pre-configured protection patterns to optimize the network protection capacity. The patterns are generated according to a fitness measure favoring patterns that provide the greatest number of useful paths at each stage of the pre-configuration design process. In the results, many final patterns resemble pre-configured cycles, then called p-cycles.

In [142], the authors propose a heuristic method for the design of survivable wavelength-routed networks with unidirectional p-cycle protection. The method is based on the efficiency ratio (ER) of each possible unity p-cycle, which is defined as the ratio of the number of working units that are actually protected by the cycle to the number of protection units of the cycle. First of all, the algorithm routes all connection demands using a shortest path algorithm and then finds the list of all possible cycles in the network topology using the algorithm from [107]. The method then calculates the efficiency score (ER) for each candidate cycle based on the working links that the cycle can protect. The heuristic algorithm iteratively selects the cycle with the highest ER. In the following, the working links that are protected by it are removed and the efficiency ratio is recalculated. The procedure is repeated until all working paths are protected. This heuristic appears to be have been also developed in [140] and [38] as well as in [26].

Lo *et al.* [73] propose a heuristic p-cycle selection approach composed of two algorithms in order to determine the p-cycle protection design. The first algorithm iteratively selects a cycle with best cycle efficiency (CE) from the set of all possible cycles until full protection is achieved. The CE is a metric involving factors such as the total protected capacity, the total spare capacity required, the idle capacity on the on-cycle links and the amount of actual protected straddling capacity. The second algorithm refines the selected cycles by removing some wasted spare capacity in order to minimize the redundancy. The numerical results show that the proposed heuristic obtains lower redundancy than CIDA [26] and achieves a gap of 3%$\sim$3.5% to the optimal solutions for two tested networks.

The approach proposed in [89] involves a combination of genetic algorithm with ILP (GA-ILP), which was used in [88]. The GA part of the approach works as a preselection algorithm for finding a small set of candidate cycles to be given as input to a final ILP model. A cycle-finding algorithm [38] enumerates subsets of eligible cycles to compose a initial population, where each subset of cycles correspond to an individual. Then, at each iteration, an auxiliary ILP problem is solved for each individual and the resulting objective function value is used to represent its fitness through the evolutionary process. The authors use this framework to solve a 200-node network obtaining a solution with redundancy of 80%. However, no optimality gap is provided.

The works summarized in this section necessarily imply one of the following major issues. On the one hand, if the extensive list of all possible cycles is enumerated and given as input to the solution method, the solution of medium, or even small, problem instances is out of reach. On the other hand, if only a subset of cycles is considered in the solution process, this may drastically compromise the quality of the solutions since there is no guarantee of optimality nor valid lower bounds.

3.1.2 Alternative solution approaches

Researchers have invested some effort to devise strategies which implicitly consider all possible cycles in order to avoid the well drawbacks of explicit enumeration of cycles.

Rajan and Atamtürk study the joint optimization of routing and p-cycle protection in [94]. They introduce a mixed-integer programming model built upon that proposed by Grover and Stamatelakis [39]. However there are some major differences concerning the assumptions considered in both works. First, in [94] the authors use unidirectional (or directed) p-cycles which is not the case in [39]. Moreover, the authors consider the following unusual assumption: the working and backup capacities on paths and cycles, respectively, are fractional values, although the total capacity installed on the links is integer. As solution approach, they use a column generation algorithm to implicitly represent all working paths and p-cycles. The pricing problems on the path variables are optimally solved using Dijkstra's algorithm [1]. The Bellman-Ford algorithm [20] is applied to heuristically generate p-cycles. It does not take straddling into account and, therefore, the reduced costs of p-cycles are estimated based on the reduced costs of the rings found using the Bellman-Ford algorithm. Hence, optimal solution of the relaxed ILP is not guaranteed. In [2], the authors pursue their study and derive valid inequalities for the model proposed in [94], using the assumption that working and protection capacities are fractional. Therein, they follow an alternative solution approach in which the pricing problem for p-cycle variables is formulated as an ILP model and solved with CPLEX. Then, a branch-and-cut algorithm is used to obtain integer solutions using only the p-cycle variables generated for the LP relaxation with column generation.

Another column generation approach for the joint design of working routing and

p-cycle protection is proposed by Stidsen and Thomadsen in [127]. They present a IP formulation, similar to the one from [42]. They propose to use the Floyd-Warshall algorithm [20] to solve the path generation subproblem. The p-cycle generation subproblem, previously referred to as the Quadratic Selective Traveling Problem in [130], is solved using the branch-and-cut algorithm described in [130]. As for the integer solutions, the ILP model with the paths and cycles collected during the column generation algorithm is solved using CPLEX. With this approach, they obtain solutions within 1% from the optimum, except for one instance.

In [108], a complex MIP model which does not require enumeration of all possible p-cycles is formulated. The model uses flow-edge variables to form cycles and takes as input the maximum number of cycles to be considered. The complexity of the model is polynomial in the number of nodes, links and cycles, which is an improvement compared to the exponential number of variables in the ILP formulation from [39]. However, the size of the formulation still grows significantly making optimal solution methods difficult for networks of medium size. Thus, the author proposes a four-step heuristic to solve the model, enabling optimization of networks with up to 25 nodes for which at most five cycles should be found.

As in [108], Wu *et al.* [134, 135] also propose ILP formulations which take as input the maximum number of cycles allowed in the solution in order to reduce the size of the problem. Although the authors claim that the formulation in [135] is based on flow conservation, both formulations actually use degree constraints [69] to determine the links and nodes composing the cycles. Compared to the work from [108], their formulations appear to be much less time consuming.

The works summarized herein are either based on column generation or depend on a parameter to restrict the size of the formulation. Remark that none of them produce optimal solutions. However, the column generation algorithm from [127] provides valid bounds for the joint optimization problem and obtains integer solutions with very small optimality gaps.

3.2 FIPP p-cycles

Failure-independent path-protecting p-cycles have not been as extensively studied in the literature as basic p-cycles. To the best of our knowledge, all published works on the topic are summarized below.

FIPP p-cycles were introduced by Kodian and Grover in [64]. The paper starts with a general background on path protection and restoration, SBPP protection scheme, and p-cycles. Then the FIPP p-cycle concept and operation are presented, together with its main advantages (see Section 2.3.3). The authors suggest two principles on which a solution approach for the design of FIPP p-cycle networks could be based. The first one consists in identifying sets of mutually disjoint working routes at an initial step, and then to define a suitable FIPP p-cycle with adequate capacity to protect each set so as that every demand is protected by at least one cycle. The second principle, in turn, consists in identifying a subset of working routes which can be properly protected by a given FIPP p-cycle which is selected from a set of candidate cycles.

Following the second principle, the authors propose an ILP model, entitled FIPP-SCP, which receives as input the set of candidate cycles as well as the working routes. The decision variables in the model correspond to the total number of unit-capacity copies of each cycle used in the solution and the number of unit-capacity copies of each cycle used to protect each demand. The notation used in the model is described bellow.

SETS

S set of links

D set of connection demands

P set of candidate cycles

DECISION VARIABLES

s_j protection capacity on link j

n^p number of unit-capacity copies of cycle p used in the solution

n_r^p number of copies of cycle p used to protect demand r

γ_r^p $\in \{0,1\}$: equal to 1 if and only if demand r is protected by cycle p

PARAMETERS

Δ	sufficiently large positive constant
∇	sufficiently small positive constant
c_j	cost of link j
d_r	number of units required by demand r
x_r^p	number of protection paths that can be provided by cycle p for demand r (equal to 0,1 or 2, depending on the relationship)
π_j^p	$\in \{0,1\}$: equal to 1 if and only if cycle p traverses link j
∂_{mn}	$\in \{0,1\}$: equal to 1 if and only if the working routes of demands m and n are not mutually disjoint

The FIPP-SCP model is given as follows:

$$\text{minimize} \quad \sum_{j \in S} c_j s_j$$

subject to:

$$\sum_{p \in P} x_r^p n_r^p \geq d_r \qquad r \in D \tag{3.1}$$

$$n^p \geq n_r^p \qquad r \in D, p \in P \tag{3.2}$$

$$\sum_{p \in P} \pi_r^p n^p \leq s_j \qquad j \in S \tag{3.3}$$

$$\nabla n_r^p \leq \gamma_r^p \qquad r \in D, p \in P \tag{3.4}$$

$$\Delta n_r^p \geq \gamma_r^p \qquad r \in D, p \in P \tag{3.5}$$

$$\partial_{mn} + \gamma_m^p + \gamma_n^p \leq 2 \qquad (m,n) \in D^2 \colon m \neq n, p \in P \tag{3.6}$$

$$\gamma_r^p \in \{0,1\} \quad r \in D, p \in P \tag{3.7}$$

$$n_r^p \in \mathbb{Z}_+ \qquad r \in D, p \in P \tag{3.8}$$

$$n^p \in \mathbb{Z}_+ \qquad p \in P \tag{3.9}$$

$$s_j \in \mathbb{Z}_+ \qquad j \in S \tag{3.10}$$

The objective function minimizes the total cost of protection capacity placed. Constraints (3.1) ensure that all demands are fully protected. Constraints (3.2) determine the correct number of required copies of each p-cycle. Constraints (3.3) ensures that sufficient protection capacity is allocated in each link. Constraints (3.4) and (3.5) determine whether p-cycle p is used to protect demand r. Constraints (3.6) play the crucial role of ensuring that any individual p-cycle only protects mutually disjoint demands. Finally, we have the domain constraints. We can notice in the model the exponential number of variables and constraints. With the solution approach used in [64], a very small set of candidate cycles should be given to the model in order to obtain likely low-quality solutions for rather small networks.

Three test networks with up to 15 nodes are used to compare the approach against SBPP, link-protecting p-cycles, and other survivability mechanisms. The results show that FIPP p-cycle designs are within 47% of SBPP solution in the worst scenario. Despite this huge difference, the large MIP gaps of the obtained FIPP solutions (up to 66%) make it very hard to arrive at a conclusion about the capacity efficiency of FIPP p-cycles in comparison with its immediate competitor SBPP.

An alternative approach based on the first principle is proposed in [66] for the non-joint design of FIPP p-cycle networks. More particularly, given the working routes for the demands, sets of mutually disjoint routes (DRS) are identified by a heuristic algorithm which works as follows. For each working route, the algorithm randomly selects working routes disjoint from the DRS under construction. The number of routes in the sets is also randomly chosen, respecting the size limit of 20. This algorithm is run ten times to make sure that each working route is included in at least ten candidate DRSs. Once the candidate DRSs are identified, ten eligible cycles are found for each set. Then, the DRSs and its assigned set of eligible cycles are provided to an ILP model, described in Chapter 4. Experimental results show that the FIPP solutions obtained with this approach are as much as 6% more capacity-efficient that conventional p-cycles and within 10-18% of the capacity used in the SBPP designs. Nevertheless, the highly heuristic nature of this approach should not be negligible.

Paper [43] reinforces the attractive advantages of FIPP p-cycles over shared protection schemes such as SBPP, in which pre-cross-connection of protection paths is not possible. All above discussed works are compiled in reference [63].

Zhang and Zhong [137] propose a novel heuristic algorithm for the non-joint optimization of FIPP p-cycles. As in [142], they consider the cycles to be unidirectional. The key idea of their algorithm is to select the most efficient cycles from a list of enumerated cycles. This resembles some of previous approaches proposed for basic p-cycles. At each iteration, an efficiency metric [142] is used to rank each cycle and the one with highest score is selected. Then, a set of pair-wise disjoint working paths is chosen to be protected by the selected cycle. This process continues until all paths have been protected. The solutions obtained with this approach are compared with their own p-cycle designs from [142]. They show that the protection capacity reduction with FIPP p-cycles is about 20% to 100% compared to link-protecting p-cycles.

Ge *et al.* [35] propose the first approach for the jointly optimized design of FIPP p-cycle networks. Their work consists of a purely heuristic algorithm without any embedded ILP component. The method is also based on the enumeration of a set of candidate cycle and DRSs for each candidate cycle. When forming DRSs, multiple working route options are provided for each demand. Then, a routine uses an efficiency metric to select the best assignments of cycles to DRSs.

A second approach for the joint optimization of FIPP p-cycles is proposed in [3]. The method is rather an extension of the DRS method from [66]. The overall strategy can be described as follows. For each demand, the N shortest routes are found, composing an eligible route set, where N is an input parameter. Then a modified version of the algorithm from [66] is used to create candidate DRSs by combining selections of route choices that are mutually disjoint into sets. Finally, all these input sets are used for the solution of an ILP formulation. The initial results obtained for a pan-european network show that the joint FIPP design is just over 1% more costly than the corresponding SBPP solution, which is an excellent result since the FIPP p-cycle solution employs only fully pre-cross-connected end-to-end path protection structures. Additional computational experiments are performed to evaluate algorithmic issues such as the number of eligible working routes and DRSs per demand.

3.3 Availability and p-cycles

Multi-failures, availability and reliability in WDM networks with p-cycles are addressed in several works such as [12, 17, 106, 107, 111]. For instance, the work in [107] focuses on the dual failures cases within a single p-cycle which cannot be survived. In order to reduce the risk of multiple failures in individual p-cycles, the physical length or the hop count of individual p-cycles must be minimized. However, minimizing the physical length or the hop count of individual p-cycle tends to increase the amount of p-cycles required to make the network survivable against single failure which in turn increases the required protection capacity. In [111], the reconfiguration of p-cycles after a link failure to protect against any subsequent link failure is investigated.

Several other approaches have been designed to improve the robustness of high capacity mesh transport networks against dual-failures. These approaches have either considered (pre-failure) strategies for addition of further protection capacity to achieve full or partial dual-failure survivability [15, 18] or have assumed reconfiguration of protection resources after the occurrence of the first failure to better withstand future failures [70, 111]. More recently, the authors of [44] have argued that, in addition to the above mentioned approaches, reductions in the physical repair time of failures (i.e., shorter outage periods) can also enhance service availability [1]. They showed that an economic strategy exists for balancing the tradeoffs between capacity investment and Mean Time To Repair (MTTR) reduction efforts to achieving high service availability in networks designed to be 100% restorable against single failures.

The authors of [16] have studied the availability in link-restorable mesh networks. The availability analysis is based on the computational analysis of the restorability of a network to all possible dual-failure scenarios and the authors explained the relation between the path availability and the service restorability. In [17], the authors developed an analytical expression for the availability of paths in networks using p-cycles as the protection mechanism. The model presented is based on the calculation of the unavailability caused by the effects of dual-failures and the authors have used the con-

1. Availability of a network is defined as "the probability of the system being found in the operating state at some time t in the future given that the system started in the operating state at time $t = 0$" [38].

cept of *cutset method* or *protection domain* to determine the service availability. Two new models were introduced for simultaneous path routing and capacity design for p-cycle networks that serve a class with higher availability requirements in addition to the traditional single-failure protected class. An availability-aware service provisioning method in p-cycle based mesh networks is presented in [85]; therein, the end-to-end service availability is analytically derived as a function of the link unavailability, using the concept of protection domain. The spare capacity is allocated, through a non-joint optimization model, to meet the availability requirement of the end-to-end traffic.

The work in [61] aimed to address some shortcomings which make the model and analysis reported in [85] inaccurate and hence to propose a more elaborate model, termed as ApC model. Concerning the availability analysis, the authors thoroughly enumerate all dual-failure scenarios which may lead to an outage on the path through which the service is routed. Then, they show that a very careful analysis must be done on each protection domain traversed by the service paths so as to avoid an overestimation of the unavailability. As a result, the proposed ApC model is more accurate but less scalable than that proposed in [85]. Therefore, the authors also propose some techniques to address the scalability issues of the ApC model, which also results in a smaller overestimation than in [85]. More recently, this availability-aware network design method has also been applied to FIPP p-cycles [96].

3.4 Further literature

p-Cycles is nowadays a hot topic in survivable network design. Many more works have been proposed addressing different issues such as dual failure, availability, demand uncertainty, and restoration time. In order to not overextend the topic with out-of-scope works, only a few references are discussed here.

Reference [122] gives theoretical arguments for the efficiency of p-cycles. For fully meshed graphs with sufficient working and spare capacity, the authors prove that p-cycles of N nodes can provide protection at best for $N(N-2)$ working links. Under the same assumptions, the authors also show that the redundancy of p-cycle networks is

$1/(d-1)$ with the number of p-cycle nodes tending to the number of network nodes, where d is the average nodal degree in the network.

In [8], Birkan *et al.* investigate robust design for a given uncertainty of the demand. A set of demand matrices which occur with a certain probability are given and the design aims to minimize a regret function. The regret function represents the tradeoff between the equipment cost of the design and the demand that is not carried because of lacking capacity in the design.

Comparative studies can be found in [49, 60]. Kennington *et al.* [60] presents a survey on the basic mathematical programming models for capacity allocation that have been proposed for mesh-based survivable networks, including link-protecting p-cycles. In [49], the authors not only overview three protection mechanisms but also perform an analytical study on the recovery time of these protection methods. They show that the time protection for configuring the cross-connects in the backup route in shared path protection is the dominant factor for restoration time, which is not needed with p-cycle protection. To compare the capacity efficiency, they revisit ILP formulations with wavelength continuity constraint. They also investigate the effect of network connectivity on the performance of capacity utilization of the methods by performing experiments on topologies with different average nodal degrees.

Researchers have also investigated cycle and path length issues in p-designs. For instance, the use of hamiltonian p-cycles in homogeneous networks is investigated in [102]. Homogeneous networks, also known as flat capacity networks, employ equal capacity in each span regardless of any future demand anticipation. Schupke *et al.* [104] present systematic results about the impact of circumference limit on the overall redundancy of a p-cycle network. In [67], Kodian *et al.* explore issues related to limiting protection path lengths in p-cycle network designs. They extend the model given in several papers [39, 46, 104] so that cycles of any length can be used, but a p-cycle cannot protect a link if its associated protection paths are longer than the hop limit required to that failure scenario. Other restrictions on candidate p-cycles have been also explored (for instance, see [111]).

Like other research works, Schupke accomplish an analysis on the resource efficiency of p-cycle networks in [109]. Nevertheless, the study differs from existing literature until then because the author addresses the physical length of working paths.

Additional works in the literature deal with further extensions of basic p-cycles, other than path-protecting p-cycles. More particularly, segment-protecting p-cycles are studied in [45, 72, 118]. Node-encircling p-cycles are proposed to protect flows transiting at every node by providing an alternate path amongst all nodes that are adjacent to the failed node [27, 124]. p-Cycle schemes have also been used for the design of protected working capacity envelopes (PWCE) [112, 113, 116, 117, 141]. The PWCE concept is a new paradigm for provisioning dynamic survivable services [40].

CHAPTER 4

A COLUMN GENERATION APPROACH FOR DESIGN OF NETWORKS USING PATH-PROTECTING P-CYCLES

4.1 Chapter presentation

This chapter presents the article entitled "A column generation approach for design of networks using path-protecting p-cycles", published in *Proceedings of the International Workshop on the Design of Reliable Communication Networks* in October 2007 and co-authored with Brigitte Jaumard, Dimitri Baloukov and Wayne D. Grover. The article concerns our investigation on a first column generation (CG) formulation for the design of failure independent path-protecting (FIPP) p-cycle survivable transport network. Previous work has proposed different formulations and heuristics for FIPP p-cycles which extend span-protecting p-cycles by adding the property of providing end-to-end failure independent path switching against either span or node failures. We develop a CG model that additionally allows the exploration of FIPP p-cycles without imposing mutual disjointness among working routes protected by the same cycle. The proposed CG model decomposes the FIPP p-cycle design problem into the master problem which takes care of the demand constraints, and the pricing problem which includes the constraints associated with the properties and the characteristics of a FIPP p-cycle. The key feature of a CG model lies in a generation of cycles motivated by the value of the reduced cost of the pricing problem, the key global indicator that is the driving element of the simplex algorithm. Results show a clear advantage of the CG model over the previous models, in particular in exploiting cycles that are not restricted to those satisfying a mutual disjointness condition on the working paths. Although it is not always possible to guarantee that the solutions obtained with the CG model are optimal, it is shown that they are very close to optimality.

4.2 Introduction

Failure-independent path-protecting (FIPP) p-cycles is a recently proposed architecture for survivable networking that extends the concept of span-protecting p-cycles [39] to allow for end-to-end path protection. The most important property inherited by FIPP p-cycles is that of complete pre-connection of protection paths. This allows FIPP p-cycles to retain the 'ring-like speed' of span-protecting p-cycles while also gaining the property of failure independence of the protection reaction for each path. In other words, when a failure occurs on any span or node, the same end-node pre-planned protection switching response takes effect. A single pre-determined and fully pre-cross-connected protection path is enabled regardless of where the failure has occurred on the working path.

The fact that the protection path is completely known and completely pre-connected before failure is beneficial from the network operator point of view because it means the backup path can be tested in a known working state before failure. This is advantageous compared to architectures such as Shared Backup Path Protection (SBPP) [115] where the protection route is known in advance, but no actual backup path is pre-connected and ready to use: It must be cross-connected on the fly immediately after failure. Pre-cross-connection means that adequate bit error rates (BER) can be assured when the protection path is substituted in real-time for the failed working signal. This property becomes even more important when considering transparent or translucent optical networks where nearly 20 [33] impairments need to be overcome before an 10Gb/s (or above) DWDM optical link can be established.

The FIPP p-cycle concept uses cyclical protection structures that can be shared by a set of working paths for protection as long as the working paths in this set are mutually disjoint or, if they are not, their protection paths are mutually disjoint. If these criteria are met, there will be no contention for spare capacity after a failure. Furthermore, the end-nodes of the working paths must also be crossed by the cycle assigned to protect them.

The original work on FIPP p-cycles, including a spare capacity placement (SCP)

ILP model, are documented in [64]. Further work, as well as the disjoint route set (DRS) method, is covered in [66]. As mentioned in these works, and in interaction among the collaborating authors on this paper, a challenge for ongoing research on the FIPP p-cycle concept was the difficulty of obtaining strictly optimal solutions for FIPP p-cycle network designs.

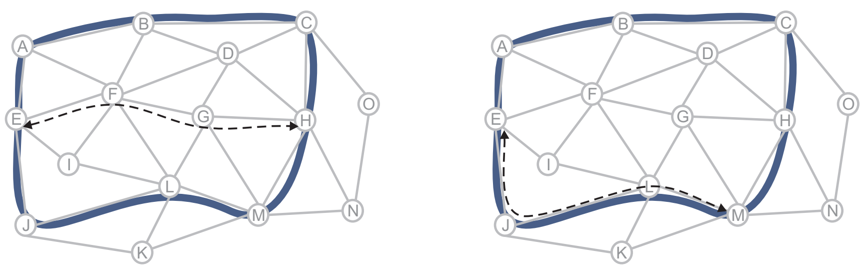
Access to optimal solutions may not be essential in practice, but it is important to advance the basic networking science in this area. We need to know, for example, the theoretical limits of how efficient FIPP p-cycle network designs can be, to understand how this architecture really relates to SBPP, say, and to be able to approach the design of good heuristics by observing the actual properties of optimal solutions for the architecture. With this in mind, a CG-based approach appears to be a promising strategy to improve the solution quality of FIPP p-cycle network designs in a reasonable amount of run time. This paper accordingly proposes a CG-based method for FIPP p-cycle network design and reports comparative performance results of the CG method and previous ILP-based solution models.

Note that the emphasis here is not on achieving a speed-up on the run times as might be important in a production-use model. Rather, the goal is to keep solution times about the same as were already being allowed or experienced with the initial approaches in [64] [66], but to achieve known-optimal terminations, or solutions with a much reduced gap against optimality.

4.3 Background and literature

The logical operations of FIPP p-cycles are explained as follows by considering the different protection relationships of a given FIPP p-cycle to working routes.

Straddling routes. A route in a pure straddling relationship to its protecting FIPP p-cycle is such that it has no span in common with the p-cycle as the example shown in Figure 4.1(a) where the working routes and cycles are represented by dashed and solid connected lines, respectively. In this case, two distinct protection paths are available on the cycle, and thus up to two working paths on this route can be protected in case of a failure. This is a direct extension of the straddler concept from span-protecting p-cycles which is one of the strongest traits contributing to their efficiency. In case of a failure, only the end-nodes of the route perform switching actions and any criteria can be adopted to assign working paths to unique protection paths. The pre-assigned direction is stored at the end-nodes where the switching action takes place as soon as a working path failure is detected. Note that the pre-defined switching action does not depend on failure type or location.



(a) (b)

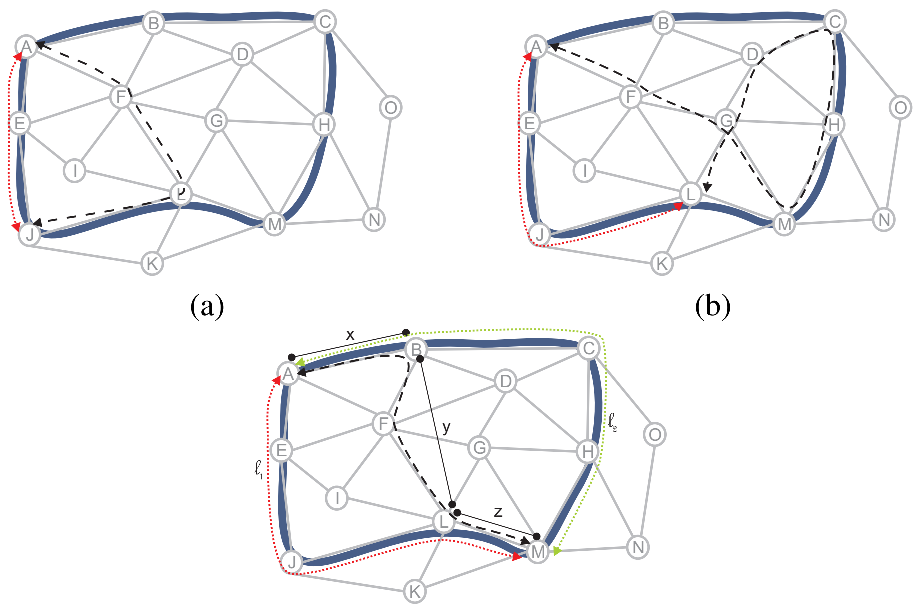
Figure 4.1: Different relationships between working routes and FIPP p-cycles: (a) fully straddling; (b) fully on-cycle.

On-cycle routes. The pure on-cycle relationship arises when all spans of the working route are crossed by the cycle protecting it as shown in Figure 4.1(b). This is the direct extension of the on-cycle concept from span-protecting p-cycles. The protection path for such a relationship is unambiguously determined as the complementary part defined by the spans of the cycle that are not shared by the working route.

Partially on-cycle routes. The partially on-cycle relationship, not faced by basic p-cycles, arises from the extension to path protection. These occur when at least one but not all spans in the working route are shared by the cycle assigned to protect it. There are two operationally different types of this relationship:

(1) The first type of partially on-cycle relationship occurs when the working path and the protection path provided by the cycle are disjoint. This is illustrated by Figure 4.2(a) and (b) where the dotted arrowed line represents the available protection path. Operationally, this is the same as the pure on-cycle relationship where

the assigned protection path is enabled regardless of where the failure affects the working path.



(a) (b) (c)

Figure 4.2: Examples of partially on-cycle routes.

(2) The second type of partially on-cycle relationship (also known as the z-case) occurs when, for a given working route, we cannot find a protection path on the p-cycle such that it does not share at least one span with its respective working route, as illustrated by the example in Figure 4.2(c). In this case, two protection paths must be considered for protection and the switching logic is performed as follows. In the example, let us set the path ℓ_1 between nodes A and M as the default pre-assigned protection path.

If segments x (A-B) or y (B-F-L) are affected by a failure, the protection path will survive and the behavior is the same as before. However, a failure on segment z (L-M) implies that the protection path defined by the default direction also fails and the affected working path must now be protected by path ℓ_2. Fortunately, this can be realized locally at the end-nodes simply by determining which side of the cycle was affected by the failure along with the working path. In case of coincidence of failure states on both working path and its pre-assigned protecting path, the surviving protection path is selected for recovery.

4.3.1 Non-disjoint working route sets

Prior work in [64] and [66] has focused on the idea of mutual disjointness between working routes protected by the same cycle. Only disjoint sets of working routes were allowed to share cycles for protection against failure. This was done to reduce the complexity of the problem and efficient results were obtained while under this assumption. However, as mentioned before, it is possible to protect non-disjoint routes using a single cycle as well, as long as the protection paths provided by the cycle are mutually disjoint from each other. This situation is illustrated in Figure 4.3(a) using the same network context as the previous examples. In the referred figure, there are two working routes, A-F-G-D-C and E-I-F-G-M, illustrated by a dotted and dashed line respectively. Although both routes go through span F-G, they can be protected by the same cycle because their protection paths (A-B-C and E-J-L-M) are mutually disjoint. Figure 4.3(b) shows an example in which two non-disjoint working routes cannot share a cycle because their protection paths cannot be disjoint. Note that the examples in Figure 4.3 only show an instance where non-disjoint straddling routes are protected by the same cycle. The idea of non-disjointness applies other route-cycle relationships as well, as long as the protection paths of the non-disjoint routes are disjoint from one another.

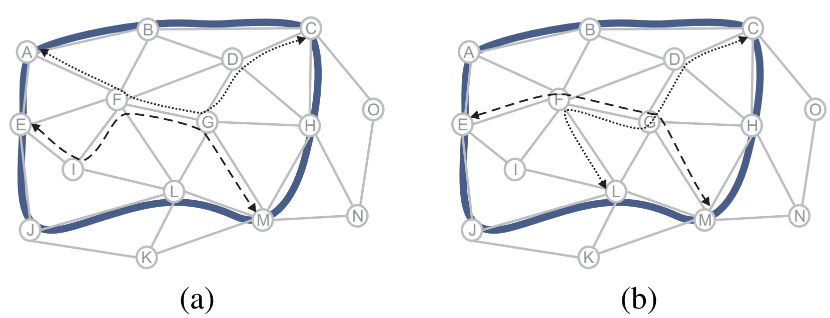


Figure 4.3: Protection of non-disjoint working paths by FIPP p-cycles.

4.4 FIPP mathematical models

4.4.1 Generalities

Three different FIPP models are introduced in the following sections: FIPP-SCP-IH, FIPP-SCP-DRS [66] and a new one, called FIPP-CG, that makes use of column generation. All correspond to a sequential optimization, i.e., working paths to be protected are defined in such a way that they are all routed over the lowest cost route between the end-nodes of each demand relation. While both FIPP-SCP models (Sections 4.5 and 4.6) assume that the set of working paths protected by a given FIPP p-cycle must be pairwise disjoint, this is no longer the case with the FIPP-CG model. The FIPP-SCP-DRS and FIPP-SCP-IH models allow for type 2 partially on-cycle relationships in the solutions while the FIPP-CG model does not, as it would entail too many variables to do so. Before going through the details of the various models, we first present a quick overview of a column generation modeling, as well as the common notation of the three models.

4.4.2 Motivation and basic theory for column generation

Column generation techniques (see, e.g., [14]) offer solution methods for linear programs with a very large number of variables where constraints can be expressed implicitly. They rely on a decomposition of the initial linear program into the *master problem* and the *pricing problem*. The master problem corresponds to a linear program subject to some explicit constraints and some implicit constraints expressed throughout properties of the coefficients of the constraint matrix. The pricing problem is defined by the optimization of the so-called reduced cost subject to the set of implicit constraints: its purpose is to help to identify and generate the most promising columns of the master problem.

The column generation solution scheme is similar to that of the simplex algorithm: It is an iterative process where, at each step, we attempt to add one or more columns to the constraint matrix of the restricted master problem (i.e., the master problem with a restricted set of variables) in order to improve the value of its objective function. The search for such columns is made through the solution of the pricing problem. If its out-

come corresponds to one or more columns with a negative reduced cost (assuming we deal with minimization), then it entails an improvement of the value of the master objective function; otherwise, if no solution of the pricing problem can be identified with a negative cost, we then conclude that the current solution is indeed optimal.
Column generation can be combined with branch-and-bound techniques for solving integer linear programs with a large number of variables, see [5] for a nice overview. Branching rules have to be devised properly in order to avoid generating a huge number of subproblems in the search tree associated with the branch-and-bound, either by branching on the variables of the master problem using cuts, or by branching on the variables of the pricing problem using classical branching schemes or cuts.

4.4.3 Notation

Consider a network represented by an undirected graph $G = (V, S)$ where V is the set of nodes and S is the set of spans, indexed by s. Let D be the set of demand relations, indexed by r, where, for each demand relation, d_r denotes the number of unit demand in the bundle and, e_r^1 and e_r^2 represent its two endpoints. Finally, let c_s be the cost of span s.

4.5 The FIPP-SCP Iterative Heuristic (IH) method

The FIPP-SCP Iterative Heuristic (IH) is included for further comparison with the new column generation approach. This method has not yet been published and is currently undergoing further investigation at TRLabs. It works by solving a subproblem, called FIPP-SCP IH Subproblem, for every cycle in the network in order to determine the most efficient working route set that can be protected by that cycle. Once this is done, the most efficient cycle (i.e., with the lowest spare capacity cost per protected working route) is recorded as being part of the solution and the working route set protected by this cycle is removed form the demand set. The FIPP-SCP IH subproblem is re-solved for every cycle in the network using the new demand set. This process continues until there are no demands left to protect. Pseudocode 4.5.1 summarizes the algorithm.

Algorithm 4.5.1 FIPP_SCP_IH Algorithm

```
while there are unprotected demands do
    for every cycle do
        Solve FIPP-SCP-IH subproblem using the remaining unprotected demands
    end for
    Choose the cycle and route set combination that result in the highest score
    Remove the demands protected by this cycle
    Record the cycle in the solution
end while
```

The following notation [1] is introduced in order to present the FIPP-SCP IH subproblem:

SETS

$D^+ =$ set of all non-zero demand relations, indexed by r.

$P =$ set of eligible cycles, indexed by p.

PARAMETERS

$c^p =$ cost of cycle p.

$$x_r^p = \begin{cases} 2, & \text{if demand } r\text{'s end-nodes are on cycle } p \text{ and} \\ & \text{its working route straddles } p; \\ 1, & \text{if demand } r\text{'s end-nodes are on cycle } p \text{ and} \\ & \text{its working route uses at least one span of } p; \\ 0, & \text{otherwise.} \end{cases}$$

$$\delta_s^r = \begin{cases} 1, & \text{if demand } r \text{ crosses span } s; \\ 0, & \text{otherwise.} \end{cases}$$

VARIABLES

$$n_r^p = \begin{cases} 1, & \text{if demand } r \text{ is protected by cycle } p; \\ 0, & \text{otherwise.} \end{cases}$$

1. The notation is redefined throughout this chapter, as in the original paper [57], in order to follow the notation used in [66]. We acknowledge that it would have been better to use a uniform notation, but the thesis's format, when presented as a collection of papers, does not allow such a change.

The FIPP-SCP IH subproblem is defined as follows: For a cycle $p \in P$,

$$\text{maximize} \sum_{r \in D^+} x_r^p n_r^p / c^p$$

$$\text{subject to:} \quad \sum_{r \in D^+} \delta_s^r n_r^p \leq 1 \qquad s \in S \tag{4.1}$$

$$n_r^p \in \{0, 1\} \qquad r \in D^+ \tag{4.2}$$

The objective function maximizes the number of demand units that the cycle protects while taking the cost of the cycle into consideration. This value represents the efficiency of the cycle. The resultant objective value is the credit score used to compare the cycles in a single iteration. The cycle with the highest credit score is the cycle that gets added to the solution. Constraints (4.1) ensure that the working routes sharing the same cycle for protection are mutually disjoint.

4.6 The FIPP-SCP DRS model

Briefly, the FIPP-SCP DRS method introduced in [66] works by first generating a large number of disjoint route sets (DRSs) of maximum specified size. A parameter controls the smallest number of DRSs that a demand is allowed to appear in. DRSs are generated by individually considering working routes and checking if they are disjoint with the working routes already in the DRS or not. If the route being considered is disjoint from the routes in the DRS, then it is added to the DRS, otherwise it is left out of the DRS. This process repeats until a given DRS size is reached. Once all the DRSs are generated, a given number of lowest cost eligible cycles are enumerated for every DRS. The eligible route set as well as the eligible cycle set are both given as input to the solver, which uses the FIPP-SCP DRS model to determine the lowest cost combination of DRSs and cycles that satisfy the specified demand.

The FIPP-SCP DRS ILP model uses the following additional notation:

SETS

A = set of eligible DRSs, indexed by a.

P_a = set of eligible cycles for DRS a, indexed by p.

PARAMETERS

$$x_r^p = \begin{cases} 2, & \text{if demand } r\text{'s end-nodes are on cycle } p \text{ and} \\ & \text{its working route straddles } p; \\ 1, & \text{if demand } r\text{'s end-nodes are on cycle } p \text{ and} \\ & \text{its working route uses at least one span of } p; \\ 0, & \text{otherwise.} \end{cases}$$

$$\delta_s^p = \begin{cases} 1, & \text{if cycle } p \text{ crosses span s;} \\ 0, & \text{otherwise.} \end{cases}$$

VARIABLES

n_a^p = number of unit-capacity copies of cycle p used as a FIPP p-cycle to protect DRS a.

The ILP formulation of the DRS-based FIPP p-cycle network design model (FIPP-SCP DRS) is as follows ([66]):

$$\min \sum_{s \in S} \sum_{p \in P} \sum_{a \in A} c_s \delta_s^p n_a^p$$

$$\text{subject to:} \sum_{a \in A} \sum_{p \in P_a} x_r^p n_a^p \geq d_r \quad r \in D \tag{4.3}$$

$$n_a^p \in \mathbb{Z}^+ \qquad a \in A, p \in P \tag{4.4}$$

The objective function minimizes the total cost of placing spare capacity in the network. Constraints (4.3) ensure that, for each demand relation r, a sufficient number of FIPP p-cycles are assigned to protect all selected DRSs of which the working route of demand r is a member.

4.7 A column generation model

We now present a column generation model, called CG model, which, although it is a formulation with an exponential number of variables, may lead to a more amenable ILP formulation for efficient solution. Each variable will correspond to a so-called cycle configuration. A *cycle configuration* C consists of a cycle p_C that satisfies a set of demand relations, that is a subset of D. It is represented by a vector $a^C = (a_r^C)_{r \in D}$ where

$a_r^C =$ number of protection units provided by cycle p_C in configuration C for demand relation $r \in D$, where $a_r^C \in \{0, 1, 2\}$ [2].

The cost of a configuration is defined by $\text{COST}_C = \sum_{s \in p_C} c_s$. Cycle configurations are not precomputed but iteratively computed using their reduced cost (i.e., the dual variables, see Section 4.7.2) in order to identify the most profitable cycle configurations, i.e., cycle and corresponding subset of demand relations.

In order to reduce the number of configurations, we introduce the concept of maximal configurations, i.e., a p-cycle configuration C is maximal if there does not exist another p-cycle configuration C' such that $a^{C'} \geq a^C$. Using maximal configurations, we may over-satisfy the demand only when doing so does not require any additional cost over the cost of satisfying the demand exactly.

A column generation always corresponds to a decomposition of the set of constraints between the master problem and the pricing problem. Here, the master problem will include the constraints that link the configurations, i.e., the demand constraints. The pricing problem will contain the constraints that are associated to a configuration and will be detailed in Section 4.7.2.

4.7.1 Master problem

The variables of the master problem, denoted by z_C, are decision variables on the configurations and are defined as follows. Let $\mathcal{C}$ be the set of all configurations. For

2. Parameters a_r^C are closely related to parameters x_r^p used in Sections 4.5 and 4.6, except that the former notation also indicates if demand relation r is protected, besides its relationship with the cycle.

$C \in \mathcal{C}$, $z_C \in \mathbb{Z}^+$ is the number of copies of configuration C used for protection.

The master problem is formulated as follows:

$$\min \sum_{C \in \mathcal{C}} \text{COST}_C z_C$$

$$\text{subject to:} \quad \sum_{C \in \mathcal{C}} a_r^C z_C \geq d_r \quad r \in D \tag{4.5}$$

$$z_C \in \mathbb{Z}_+ \qquad C \in \mathcal{C}. \tag{4.6}$$

Constraints (4.5) are the demand constraints where $a_r^C z_C$ gives the units of demand relation r protected by p_C .

4.7.2 Pricing Problem

By definition, the pricing problem corresponds to the optimization problem of minimizing the reduced cost (with respect to linear programming definition) subject to the constraints that must be satisfied by a given configuration, which are: definition of a cycle, identification of the demand that can be protected by the cycle, prohibition for a span to be used as a working and a protection span at the same time for the same demand.

Additional notation for the mathematical formulation of the pricing problem is given as follows:

The mathematical formulation of the pricing problem is given as follows. Let us first give the expression of its objective function corresponding to the reduced cost of a column of the master problem, i.e., the cost of the cycle less the prices of the protected demands.

$$\overline{\text{COST}}_C = \text{COST}_C - \sum_{r \in D} a_r^C u_r = \overbrace{\sum_{s \in S} x_s c_s}^{\text{COST}_C} - \sum_{r \in D} \overbrace{\sum_{s \in \omega(e_r^1)} x_s^r}^{a_r^C} u_r.$$

SETS

$\omega(v)$ $= \left\{ \{v, j\} : \{v, j\} \in S \right\}$: Set of all spans adjacent to node $v \in V$.

$\omega(V')$ $= \left\{ \{i, j\} : i \in V', j \in V \setminus V' \right\}$: Set of all spans whose one end-node belongs to V' and the other does not, for $V' \subset V$.

$S(V')$ $= \left\{ \{i, j\} : i, j \in V' \right\}$: Set of all spans whose end-nodes belong to V', $V' \subset V$.

PARAMETERS

u_r = dual prices

c_s = cost of span s

$$\rho^{rr'} = \begin{cases} 1, & \text{if working paths of demands } r \text{ and } r' \text{ are non-disjoint;} \\ 0, & \text{otherwise.} \end{cases}$$

$$\delta_s^r = \begin{cases} 1, & \text{if working path of demand } r \text{ uses span } s\text{;} \\ 0, & \text{otherwise.} \end{cases}$$

VARIABLES

$$x_s = \begin{cases} 1, & \text{if the cycle uses span } s\text{;} \\ 0, & \text{otherwise.} \end{cases}$$

$$x_s^r = \begin{cases} 1, & \text{if any protection path for demand } r \text{ uses span } s\text{;} \\ 0, & \text{otherwise.} \end{cases}$$

Let us now express the set of constraints of the pricing problem.

$$\min \overline{\text{COST}}_C$$

subject to:

$$\sum_{s \in \omega(v)} x_s \leq 2 \qquad v \in V \tag{4.7}$$

$$\sum_{s' \in \omega(v): s' \neq s} x_{s'} \geq x_s \qquad v \in V, s \in \omega(v) \tag{4.8}$$

$$\sum_{s \in \omega(V')} x_s \geq x_{s'} + x_{s''} - 1 \quad V' \subset V, s' \in S(V'), s'' \notin S(V) \tag{4.9}$$

$$x_s \geq x_s^r \qquad s \in S, r \in D \tag{4.10}$$

$$\sum_{s \in \omega(e_r^1)} x_s^r = \sum_{s \in \omega(e_r^2)} x_s^r \qquad r \in D \tag{4.11}$$

$$\sum_{s \in \omega(v)} x_s^r \leq 2 \qquad r \in D, v \in V \tag{4.12}$$

$$\sum_{s' \in \omega(v) | s' \neq s} x_{s'}^r \geq x_s^r \qquad r \in D, v \in V \backslash \{e_r^1, e_r^2\}, s \in \omega(v) \tag{4.13}$$

$$2 - x_s^r - x_s^{r'} \geq \rho^{rr'} \qquad s \in S, (r, r') \in D^2 : r \neq r' \tag{4.14}$$

$$x_s^r \leq 1 - \delta_s^r \qquad s \in S, r \in D \tag{4.15}$$

$$x_s^r \in \{0, 1\} \qquad s \in S, r \in D \tag{4.16}$$

$$x_s \in \{0, 1\} \qquad s \in S \tag{4.17}$$

The first three sets of constraints take care of the construction of the cycle. Constraints (4.7) and (4.8) ensure flow circulation on the cycle variables, i.e., they address the cycle construction by stating that, at all nodes, the number of incoming/outgoing flows must be either 0 (p-cycle does not go through node v) or 2 (p-cycle contains node v and two adjacent spans such that $x_s = 1, s \in \omega(v)$). Constraints (4.9) prevent subcycles, i.e., p-cycle made of more than one cycle. Those constraints are a variant of the classical subcycle elimination constraints of the Traveling Salesman Problem (TSP), with the difference that a p-cycle does not necessarily include all nodes while a TSP tour must include all nodes exactly once. A drawback of this formulation is that there is an exponential number of subcycle elimination constraints. Notice, however, that the pricing problem does not need to be solved exactly at each iteration, and if one is able to design an efficient heuristic to solve it, we may need to solve exactly the pricing problem only once to confirm that the heuristic has indeed found the optimal solution.

The following sets of constraints will define the protection path. Constraints (4.10) establish the relationship between the set of variables associated with the cycle, and those associated with the protection paths. The next three sets of constraints correspond to the definition of a protection path. These constraints address the flow between the two end-nodes of a demand relation and ensure that the amount of flow is equal to the protection amount (it can be only a fraction of the number of unit demand of the demand relation) provided by the cycle under construction. Constraints (4.11) state that the flow exchanged between the end-nodes must be equal, i.e., a protection path must end at both end-nodes of a demand. At the end-node, constraints (4.12) allow at most two protection paths for each demand relation. At intermediate nodes, constraints (4.12) and (4.13) together ensure flow conservation, i.e., the number of incoming/outgoing flows must be either 0 (protection path of demand relation r does not use node v) or 2 (protection path contains node v and two adjacent spans such that $x_s^r = 1, s \in \omega(v)$).

The last two sets of constraints take care of the exclusive use of a given span in either working or protection path. Constraints (4.14) state that non-disjointly routed demands cannot share a span for their protection. Constraints (4.15) prevents from using a given span in both working and protection paths of each demand relation.

4.7.3 Solving the CG Model

In order to solve the CG model, we propose to use a branch-and-bound method assuming that the linear relaxation of the master problem is solved using column generation techniques. Let us first address the issue of solving the linear programming relaxation of the master problem, i.e., the problem described in 4.7.1 with constraints (4.6) relaxed to

$$z_C \geq 0 \qquad C \in \mathcal{C}.$$

The first difficulty lies in the solution of the pricing problem. As we mentioned before, the pricing problem does not need to be solved exactly at each iteration of the column generation process, as long as we are able to find a configuration with a negative reduced cost in order to be able to iterate. The pricing problem can be described as a multi-commodity flow problem with side constraints corresponding to the definition of a cycle (that is usually not a Hamiltonian cycle) on which the flows circulate. Constraints (4.9) that prevent from considering sub-cycles are very costly as there are an exponential number of them. In order to overcome this difficulty, we introduce them only as needed following the principle of the "lazy constraints" feature of CPLEX™: It means that the pricing problem is solved iteratively, starting with no (or a very small number of) sub-cycle constraints, and adding some sub-cycle constraints that are violated until reaching a feasible solution which satisfies all (implicit or explicit) sub-cycle constraints. In order to save computing time, we also stop the solution of the pricing problem as soon as we obtain a feasible solution (i.e., a configuration) with a negative reduced cost. Therefore, in the iterative process with the lazy constraint like feature, the final solution is the first found feasible solution with a negative reduced cost. In practice, we need to introduce a very small number of sub-cycle constraints, and therefore the computing cost of iterative process which leads to possibly several solutions of the pricing problem is counter balanced by the smaller sizes of the pricing problems that need to be solved. Indeed, in practice, we very rarely needed to solve the pricing problem more than twice. Note that we never eliminate any sub-cycle constraints that have been introduced. Consequently, their number increases as we go further down in the branch-and-bound search tree. Although we could eliminate some of them, it offers the advantages that in practice, most of the time, at each iteration, the solution of the first pricing problem is very often feasible, i.e., satisfies all sub-cycle constraints even if only a small number of them have been explicitly introduced in the pricing problem.

It was observed that the number of columns generated until the optimality condition of the linear programming relaxation of the master problem was satisfied (i.e., no more column with a negative reduced cost) was rather limited (see the details in Section 4.8). For this reason, we did not develop any custom branch-and-bound (B&B) to get an integer solution for the master problem, but instead we used the CPLEX™ B&B to solve the ILP restricted master problem with the set of columns generated for the exact solution of its linear programming solution. Although we cannot claim that we have obtained optimal solutions for the CG model, the resulting solutions are already quite satisfactory in comparison with those obtained with previous models, as it is shown in Section 4.8.

4.8 Computational Results

The solutions for the FIPP CG, FIPP SCP IH and FIPP SCP DRS design models were obtained by implementing the prior model in C++ and the latter two models in AMPL 9.0. The FIPP SCP IH and FIPP SCP DRS models were solved using CPLEX 9.0 MIP solver and the results obtained are based on complete terminations with a MIPGAP of 0.01. The FIPP CG model was solved using CPLEX 10.0 MIP solver. The models were run using the COST239 European network [6], containing 11 nodes and 26 spans, as well as a 15-node family of networks whose number of spans ranges from 16 to 30 [25]. The number of demand relations is 55 for the COST239 instance, and 105 for all instances of the 15-node family.

For the FIPP SCP DRS solutions provided in this paper, 30 DRSs were generated per demand in the network. This means that every demand appeared in at least 30 different DRSs. For every DRS, the three lowest cost cycles eligible to protect that DRS were enumerated and added as input to the problem. The maximum number of working routes per DRS was set to 12. Additionally, to remain true to the method used in [66], a single route DRS was generated for every demand to counteract the effects of any strong forcer demands. In other words this was done to prevent any particularly high demand from forcing the solver to place additional large cycles where it could instead use a small dedicated cycle to protect that demand. The FIPP SCP IH solutions were obtained by running the FIPP SCP IH algorithm where all possible cycles in the network were considered as eligible candidates.

The FIPP-CG method uses the results obtained by FIPP-SCP IH heuristic as starting feasible solutions. The results for this method are summarized in Table 4.I. The table reports the final cost of spare capacity as well as the gap against optimality, i.e., the gap between the optimal solution of the linear relaxation of the master problem and the best known solution. In some cases, where the method reached a running time limit of 5 hours for solving the linear relaxation, this gap is only an under-estimation (values followed by $^-$). We also provide the total number of unit cycles used in the final solution, i.e., $\sum_{C \in \mathcal{C}} z_C$, along with the number of distinct cycles used (of which there could be more than one instance of). The average length of the cycles in the final solution varies between 9.5 and 11.6 spans for the 15-node family. In the following two columns, we indicate the overall number of generated configurations by distinguishing between the number of null and positive master variables (z_C). The second last column contains the number of configurations which take advantage of non-disjointness. We observe that about half of the configurations are associated with non pairwise disjoint working paths. Finally, the last column contains the overall number of generated cycles (the number in parenthesis corresponds to the distinct number of cycles in the initial solution). Note that the initial set of columns provided to the CG model (in parenthesis in the last column) is composed of the cycles and configurations deduced from the best solution of the FIPP-SCP IH model. Although using these solutions has no impact on the solution of the CG model (assuming that no time limit is set), it certainly speeds up the convergence, allowing the generation of more meaningful values for the dual variables, hence the generation of better configurations. In terms of the number of configurations in the best solution (as indicated by the number of configurations such that $z_C > 0$), it is usually rather close to the number of cycles except for the first instances of the 15-node family as the number of eligible cycles is very limited for the first three instances. Although we can certainly look at an enhanced, more compact column generation model where each configuration would be associated to a cycle and vice-versa, it will not significantly reduce the number of generated columns.

The results for FIPP SCP DRS and FIPP SCP IH are documented in Tables 4.II and 4.III, respectively. These tables contain the final design costs along with the number of unit cycles as well as the number of distinct cycles used in these solutions. Furthermore, the relative cost difference to the CG solutions is also reported in the last column of these tables. Note that Table 4.II only contains the results for the 15m30s network family up to the 23rd span instance. It was at this point that the solver did not return with a solution within the specified MIPGAP due to the large increase in the number of constraints due to the large number of parameters used.

One of the weaknesses of the DRS method is that it does not scale very well with the increase in network size. The more DRS/cycle combinations there are to consider, the harder it becomes to get the DRS method to solve the problem to completion. It is possible to reduce the number of parameters, but this severely degrades the quality of the solution that the solver arrives at. The DRS method is capable of reaching the optimal solution, but only if all possible combinations of cycles and DRSs are given to it as input. This cannot be done for anything larger than the simplest network because of the huge number of DRS/cycles possible even for a medium sized network. Therefore whenever parameters are introduced, the DRS method does not yield an optimal solution, in general, but only the optimal combination of the DRSs and cycles given to it as input.

As an alternative to the FIPP SCP DRS method, the iterative heuristic method was introduced. It was possible to get solutions for the entire 15n30s network family using this method as documented in Table 4.III.

It can be observed that the spare capacity cost of the CG method is lower than both DRS and IH method solutions and that the DRS and the IH methods are relatively close to each other in cost. The cost improvement when comparing the DRS and IH results to the FIPP CG solutions is as much as 19% for the 15n30s and 37% for the cost 239 network. Also it can be seen that the relative difference to the CG solutions increases as the number of spans is increased from 16 to 30 in the 15n30s family of networks. Taking into account that around half of the configurations in the final solution of the CG method take advantage of non disjoint working paths, it is unclear at this point whether the improved solutions are due to this effect or to the global search scheme entitled by the column generation method.

The benefit of the CG method is twofold: it is able to consider a more general view of the route sets that can be protected while also being able to arrive at nearly optimal solutions by being able to consider only the configurations that reduce the objective value of the master problem. The DRS method considers not only a smaller range of route combinations, but also only a small subset of all possible combinations, and thus it is not able to achieve the efficiencies of the CG method. Likewise, the IH method also only considers route combinations that are disjoint from one another and, while it is able to optimally determine the best disjoint route combination for a given cycle that maximizes the credit score, it is unable to consider the problem as a whole and thus does not gain any benefit from the interdependencies of the cycles and their protected sets.

4.9 Conclusion

We have investigated a first column generation (CG) model for the efficient design of FIPP p-cycles, and successfully compared it against the FIPP-CSP DRS method as well as the FIPP-SCP IH heuristic. While previous FIPP-SCP methods work under the assumption that working paths are pairwise disjoint, this is not the case for the FIPP-CG model. This, in addition to the fact that the CG method is capable of generating near optimal solutions, can provide us with insight into what a really good FIPP solution actually looks like and can help with the conception and improvement of heuristic methods. Despite the quality of the solutions presented, there is still room for improvement of the CG model. In particular, the solution method may be sped up if the pricing problem is solved by a heuristic able to generate configurations with negative reduced cost. Additionally, a more compact/exact CG model can still be designed.

Table 4.I: Results obtained by the FIPP-CG Algorithm

Problem Instances	Cost	Gap (%)	# Unit Cycles	# Distinct Cycles	Overall # Configurations $z_C > 0$	Overall # Configurations $z_C = 0$	# Non-disjoint Configurations	Overall # Cycles
COST239	68840	2.5	17	13	15	534	6	286 (20)
15n30s1-16s	387958	0.0	185	3	73	169	0	3 (3)
15n30s1-17s	286901	0.2	140	6	72	217	18	7 (6)
15n30s1-18s	266080	0.2	126	7	60	350	22	11 (10)
15n30s1-19s	229854	0.1	108	11	59	338	20	21 (16)
15n30s1-20s	217963	0.1	108	13	64	472	22	31 (23)
15n30s1-21s	192269	0.4	101	17	63	412	26	50 (27)
15n30s1-22s	182550	0.6	101	19	66	411	30	72 (34)
15n30s1-23s	177432	0.5	100	27	61	505	32	111 (45)
15n30s1-24s	175990	0.6^-	100	29	58	636	23	178 (45)
15n30s1-25s	157140	0.9^-	92	36	56	504	21	196 (43)
15n30s1-26s	106526	6.1^-	55	39	45	395	18	287 (39)
15n30s1-27s	117220	5.5^-	61	41	45	399	28	287 (51)
15n30s1-28s	114055	4.5^-	59	44	46	333	23	303 (44)
15n30s1-29s	106449	2.6^-	56	33	42	534	28	345 (39)
15n30s1-30s	110738	1.0^-	62	40	51	473	34	326 (38)

Table 4.II: FIPP-SCP DRS Results

Problem Instances	Cost of spare	# Unit Cycles	# Distinct Cycles	Cost FIPP-CG (%)
COST239	93345	24	17	36 %
15n30s1-16s	393094	189	3	1 %
15n30s1-17s	305065	146	6	6 %
15n30s1-18s	272661	127	8	2 %
15n30s1-19s	246035	110	12	7 %
15n30s1-20s	239942	113	17	10 %
15n30s1-21s	208501	101	26	8 %
15n30s1-22s	202771	101	40	11 %
15n30s1- $\geq$23s	-	-	-	-

Table 4.III: FIPP-SCP IH Results

Problem Instances	Cost of spare	# Unit Cycles	# Distinct Cycles	Cost FIPP-CG (%)
COST239	94095	25	20	37 %
15n30s1-16s	396935	192	3	2 %
15n30s1-17s	310094	152	6	8 %
15n30s1-18s	291047	139	10	9 %
15n30s1-19s	259825	129	16	13 %
15n30s1-20s	246505	124	23	13 %
15n30s1-21s	215708	114	27	12 %
15n30s1-22s	198789	104	34	9 %
15n30s1-23s	199378	106	45	12 %
15n30s1-24s	201207	107	45	14 %
15n30s1-25s	179897	100	43	14 %
15n30s1-26s	115848	60	39	9 %
15n30s1-27s	137887	74	51	18 %
15n30s1-28s	128957	71	44	13 %
15n30s1-29s	126745	73	39	19 %
15n30s1-30s	132294	79	38	19 %

CHAPTER 5

A UNIFIED FRAMEWORK FOR SHARED PROTECTION SCHEMES IN OPTICAL MESH NETWORKS

5.1 Chapter presentation

In the following, we present the article entitled “A column generation approach for shared protection schemes in WDM mesh networks” co-authored with Brigitte Jaumard. This article was accepted for publication in the Brazilian journal of operations research, *Pesquisa Operacional*, in July 2009. A preliminary version of this article was published under the title of “Revisiting p-cycles / FIPP p-cycles vs. shared link / path Protection” in *Proceedings of the 17th International Conference on Computer Communications and Networks (ICCCN)*, August 2008.

Herein, we propose a further investigation on the bandwidth protection costs of p-cycles and FIPP p-cycles in comparison with those of shared link and path protection by applying column generation technique to solve relaxed LP models for the four protection schemes, and then solving the resulting ILP models. Provably near-optimal solutions allow us to perform accurate quantitative comparisons on real-world networks.

5.2 Introduction

A key driver for optical networking technology has been the sustainment of the Internet growth. Researchers have contributed with many advances in optical wavelength division multiplexing (WDM) equipment and networking architectures to meet these Internet traffic needs, leading to an optical technology that currently offers immense bandwidth scalability. As it has gained notable market traction over the last several years, enterprises rely more and more on their communication services to reach their business milestones. Given the immense scale of WDM networks and how much downtime can cost a business, service survivability issues are of paramount importance. In spite of an average of 3 to 13 cuts for every 1000 miles of fiber in long haul networks [38], failures have a strong impact on direct revenues.

The design of a mesh WDM network usually proceeds in two steps, firstly the establishment of the *working* (or routing) *paths* with the objective of minimizing the working capacity or the equipment cost, among others. Secondly, *protection paths* are set in order to offer resilience against failures. It is well known that fiber cuts are the dominant failure pattern, and that protection against single link failure is a reasonable assumption. Although some researchers have started to investigate the joint design of working and protection paths, our focus here is on the sequential approach, in which working paths are defined a priori, usually via shortest routes.

Several survivability strategies can be found in the literature, all based on a bunch of features that has an impact on the network operation and/or design [38]. Firstly, a survivable network can either use a protection or a restoration scheme. In a *protection scheme*, the redundant resources are precomputed and reserved in advance. On the opposite, *restoration schemes* take action in real time, including resource allocation and path cross-connections, based on the failure and the state of the network at the time of failure. While restoration schemes are usually more bandwidth efficient because they do not allocate spare capacity in advance, protection schemes have faster restoration time and can always guarantee recovery from failure. Secondly, one can choose either link or path protection. *Link protection/restoration* consists in protecting each link as one entity, regardless of the connection demands that go through it, while *path protection/restoration* protects each demand individually by providing a surviving protection path between its end nodes. Although path protection (restoration) schemes lead to an efficient utilization of backup resources, they also lead to a longer failure detection and recovery than link protection (restoration). Moreover, survivability mechanisms can use either *dedicated capacity*, where spare capacity for each link or path is exclusively allocated, or *shared capacity*, where spare capacity can be shared among several protection paths under the single failure assumption.

In this work, we propose to investigate the bandwidth protection costs of the classical shared link and path protection schemes against two well-known particular cases: p-cycles [39] and failure-independent path-protecting (FIPP) p-cycles [64]. The key advantage of pre-configured protection cycles, or p-cycles for short, lies in their switching speed and simplicity, similar to ring networks, as the protection paths around the surviving portions of the cycle are pre-connected at the outset and the only required switching actions take place at the end nodes of the failure. In spite of many existing studies on protection/restoration schemes [60, 133, 139] and although it is known that p-cycles are less capacity efficient, there has been no systematic analysis of how much bandwidth p-cycles schemes require in comparison with the basic shared link and path protection. Due to the highly combinatorial nature of p-cycle designs, nearly all studies are based on an explicit enumeration of cycles, resulting in difficulties for assessing the quality of the solutions provided by the resulting huge ILP models. Therefore, our goal is to accomplish an accurate comparison based on results provably close to optimality by using a specialized tool for solving large scale mathematical programs: the column generation method. A unified mathematical programming framework is presented to minimize the overall protection cost of those protection schemes.

Column generation technique (see, e.g., [14, 24, 74, 76, 77] for a reference on network design) is a powerful tool for solving linear programs with a very large number of variables where constraints can be expressed implicitly. The method relies on a decomposition of the initial linear program into a *master problem* and a *pricing problem*. The master problem corresponds to a linear program subject to a first set of explicit constraints and a second set of implicit constraints expressed throughout properties of the coefficients of the constraint matrix. The pricing problem consists in the optimization of the so-called reduced cost subject to the set of implicit constraints: It either identifies favorable columns to be added to the master problem or indicates that no such column exists.

The paper is organized as follows. The next section presents an ILP model for each protection scheme along with its respective column generation algorithm. Experimental results are depicted in Section 5.4, followed by the conclusion of the study.

5.3 Mathematical models for network protection

Consider a WDM mesh network represented by a graph $G = (V, E)$ where V is the set of nodes and E is the set of bidirectional links, indexed by e in a non-failure state and by f in a failure state. Each link e is associated with a cost c_e. Let K be a set of connection requests, where each request k requires b_k units of working traffic (channels) between nodes o_k and d_k. We assume that the working route of each connection request is given and denoted by WP_k. The overall working traffic on a link e is denoted by $w_e = \sum_{k \in K:\, e \in \text{WP}_k} b_k$. In this section, we present mathematical models, suitable to be solved by a column generation method, for establishing protection paths with enough capacity to protect all working traffic while minimizing the overall protection cost. Although the working traffic is non-bifurcated, it is assumed that it can be split into integer fractional parts (integer numbers of channels) during protection. For instance, the working traffic of a given request can be restored on different protection routes, each carrying an integer fraction of the failed traffic.

5.3.1 Link protection

Under a link protection scheme, the interrupted traffic is rerouted only around the failed link. Thus, the total amount of working traffic on each link is considered for protection, regardless of the connections going through it. An example of the classical shared protection scheme is illustrated in Figure 5.1(b).

5.3.1.1 Shared link protection model

Let $\mathcal{P}_f^e$ be the overall set of potential protection paths for link f that use link e, and $\mathcal{P}_f = \bigcup_{e \in E} \mathcal{P}_f^e$, both indexed by p. The variables of the model represent the amount of spare capacity allocated (number of spare channels) on each link e, denoted by s_e, and the amount of traffic restored through each path p protecting link f, denoted by n_f^p. The Shared Link Protection Model (SLP-M) is defined as follows:

$$\min \sum_{e \in E} c_e s_e$$

$$\text{subject to:} \quad \sum_{p \in \mathcal{P}_f} n_f^p \geq w_f \qquad \forall f \in E \tag{5.1}$$

$$\sum_{p \in \mathcal{P}_f^e} n_f^p \leq s_e \qquad \forall e \in E, f \in E \setminus \{e\} \tag{5.2}$$

$$n_f^p \in \mathbb{Z}_+ \qquad \forall f \in E, p \in \mathcal{P}_f \tag{5.3}$$

$$s_e \in \mathbb{Z}_+ \qquad \forall e \in E. \tag{5.4}$$

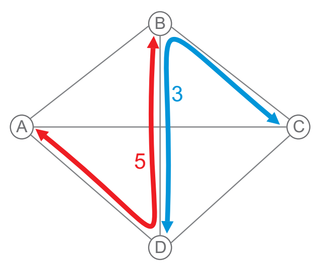
The objective function minimizes the overall protection cost. The link costs c_e can represent information such as equipment cost at end nodes, link length, etc. If $c_e = 1, \forall e \in E$, the total protection capacity is minimized. Constraints (5.1) ensure that the overall working traffic on each link is protected. Constraints (5.2) determine the total number of spare channels that must be allocated on each link.

This model has a huge number of variables n_f^p as there is an exponential number of potential protection paths for each link. To avoid explicit representation of all paths, a column generation approach is applied so that a solution of the LP-relaxed SLP-M model (SLP-RM) can be obtained by generating only a very small fraction of the potential paths without preventing from reaching the optimal solution. The column generation algorithm initially solves a restricted SLP-RM, which contains only a small subset of paths, and generates additional paths when needed. At each iteration, the pricing problem is set and solved for a given link f, associated with variable n_f^p: It consists in finding a path $p \in \mathcal{P}_f$ with minimum reduced cost. Each link is considered in a round robin fashion. If a path with negative reduced cost is found, the solution of the restricted SLP-RM can be further improved and the path is added to the restricted constraint matrix. Otherwise, the solution of the restricted SLP-RM is also the solution of SLP-RM and, consequently, we can claim that the LP relaxation of SLP-M has been solved to optimality.

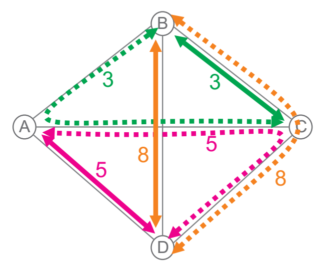
For each link f, the pricing problem consists in minimizing the reduced cost of variables n_f^p, denoted as follows:

$$\bar{c}_f^p = -\mu_f - \sum_{e \in p} \pi_f^e,$$

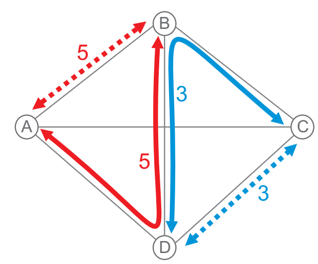
where $\mu_f \geq 0$ and $\pi_f^e \leq 0$ are the dual prices of constraints (5.1) and (5.2), respectively. For a given link f, the term μ_f is constant, and therefore the path with the minimum reduced cost can be found as the shortest path in the modified graph $G' = (V, E')$, where $E' = E \backslash \{f\}$ and the link costs are defined by $c'_e = -\pi_f^e$.



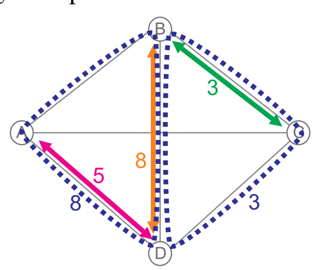
(a) Two working paths, with their respective amount of traffic, which are used in the following examples.



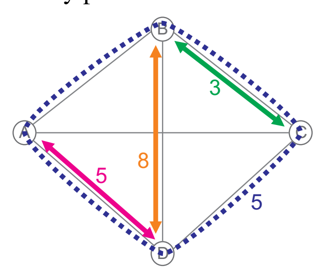
(b) Classical link shared protection. The necessary spare capacity on link A-C is 5 because it is shared by paths A-D and B-C.



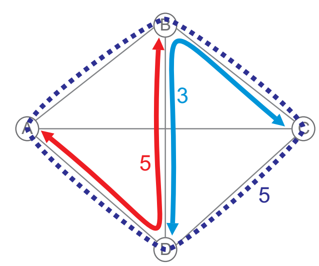
(c) Classical shared path protection.



(d) Two p-cycles for link protection.



(e) Here, only one p-cycle protects the same traffic as before. As link B-D straddles the cycle, the necessary spare capacity on this cycle is 5 .



(f) A FIPP p-cycle.

Figure 5.1: Examples of the four protection schemes. Working and protection paths are represented by solid and dashed lines respectively, and the numbers represent the amount of traffic.

5.3.1.2 p-Cycle model

Introduced in [39], p-Cycles are fully pre-connected cyclic protection structures with pre-planned spare capacity. When a link failure occurs, only the two end nodes of the failed span perform protection switching. Unlike rings, p-cycles protect against straddling link failures, enabling two protection paths, one on each half of the cycle, with only one unit of spare capacity. p-Cycles also provide protection against failures on links over the ring itself, as illustrated by Figures 5.1(d) and 5.1(e).

The following notation is introduced. Let $\mathcal{P}$ be the overall set of candidate p-cycles, indexed by p. The decision variables are denoted by n_p and represent the amount of traffic circulating through p-cycle p , i.e., the number of unit copies of p-cycle p. The coefficients $a_f^p \in \{0, 1, 2\}$ define the protection provided by p-cycle p for link f: 1 if link f lies on cycle p; 2 if link f straddles cycle p (f is a chord of the cycle); and 0 otherwise.

The p-cycle model (PC-M) for link protection can be written as follows:

$$\min \sum_{p \in \mathcal{P}} \text{COST}_p n_p$$

$$\text{subject to:} \quad \sum_{p \in \mathcal{P}} a_f^p n_p \geq b_f \qquad f \in E \tag{5.5}$$

$$n_p \in \mathbb{Z}_+ \qquad p \in \mathcal{P} \tag{5.6}$$

The objective function calculates the overall protection cost, where COST^p is the unit cost of p-cycle p, given by $\text{COST}^p = \sum_{e \in p} c_e$. Constraints (5.5) ensure that the working traffic on each link is fully protected.

Because we also aim to solve the linear relaxation of the model above with on-line generation of the cycles, by using a column generation technique, we need to define the formulation of the pricing problem. The pricing problem to be solved at each iteration of the column generation algorithm looks for the lowest cost cycle protecting the most profitable links. It corresponds to an NP-hard optimization problem, the Quadratic Selective

Traveling Salesman Problem, described in detail and solved using a branch-and-cut algorithm in [130]. Let us introduce three sets of variables. The first one contains variables x_e such that $x_e = 1$ if e defines one of the links supporting the sought cycle, and 0 otherwise. The second set is made of variables y_v such that $y_v = 1$ if v belongs to the cycle, and 0 otherwise. Last, the third set is made of variables z_e such that $z_e = 1$ if link e is protected by the cycle, and 0 otherwise. Constants $\mu_e \geq 0$ are the dual prices from constraints (5.5) of master problem. The pricing problem can be formulated as follows:

$$\min \quad \sum_{e \in E} (c_e + \mu_e)\, x_e - 2 \sum_{e \in E} \mu_e\, z_e$$

subject to:

$$\sum_{e \in \omega(v)} x_e = 2\, y_v \qquad v \in V \tag{5.7}$$

$$z_e \leq y_v \qquad v \in V,\ e \in \omega(v) \tag{5.8}$$

$$z_e \geq y_v + y_{v'} - 1 \qquad v, v' \in V,\ e = \{v, v'\} \in E \tag{5.9}$$

$$\sum_{e \in \omega(V')} x_e \geq 2\,(y_v + y_{v'} - 1) \qquad V' \subset V,\ 3 \leq |V'| \leq |V| - 3,\ v \in V',\ v' \in V \setminus V' \tag{5.10}$$

$$y_v \in \{0, 1\} \qquad v \in V \tag{5.11}$$

$$z_e, x_e \in \{0, 1\} \qquad e \in E \tag{5.12}$$

The objective function is composed of two terms: one corresponding to the compound cost of the cycle under construction, and another one associated with the reward obtained from the protection provided to the links. Constraints (5.7) establish conditions to build cycles. Constraints (5.8) and (5.9) link y and z variables. Constraints (5.10) are the subtour elimination constraints, which ensure to build only one cycle at a time, as building more than one cycle would entail difficulty to identify the straddling links. And finally, we have binary domain constraints. Note that the domain of z variables can be relaxed to [0,1] as they correspond to the multiplication of y variables.

The PC-M model can be seen as an adaptation of that proposed in [127]. In there, the authors present a column generation model for the joint optimization of working routing and p-cycle protection, where two pricing problems are solved at each iteration: One for generating candidate working paths and another one for generating protecting cycles. Here, only protection is considered as we investigate the sequential optimization approach. In such a case, the pricing problem for generating p-cycles is the same as in [127].

5.3.2 Path protection

Path protection consists in protecting each demand individually by providing a protection path, link disjointly routed from its working path. In case of a link failure, a notification signal is sent to the end nodes of each connection traversing the failed link in order for them to switch the traffic over from the working path to the protection path. We discuss two failure-independent schemes: the Shared Backup Path Protection (SBPP) and the Failure-Independent Path-Protecting (FIPP) p-cycles, illustrated by Figures 5.1(c) and 5.1(f) respectively.

5.3.2.1 Shared path protection model

Known in the literature as Shared Backup Path Protection (SBPP) [38] and Global Backup Path Protection [11], this is a failure independent path protection scheme where the traffic on an affected working path is switched to a predefined and disjointly routed protection path. Cross-connection operations to set the protection paths are performed at the time of the failure. Unlike 1+1 protection, SBPP allows the spare capacity allocated to protection paths to be shared over failure-disjoint working paths.

Let us introduce the model for the optimal design of SBPP. Here, the set $\mathcal{P}_k^e$ is defined for each connection request k as the set of candidate protection paths for k using link e and the set $\mathcal{P}_k$ is the set of all candidate protection paths for k, denoted by $\mathcal{P}_k = \bigcup_{e \in E} \mathcal{P}_k^e$. The variables are denoted by s_e, representing the number of spare channels allocated on link e, and by n_k^p, representing the amount of restored traffic through the path p for protecting demand k.

The optimal design of networks using SBPP as protection scheme is defined by the model SPP-M below:

$$\min \sum_{e \in E} c_e s_e$$

$$\begin{aligned} \text{subject to:} \quad & \sum_{p \in \mathcal{P}_k} n_k^p \geq b_k & k \in K \quad & (5.13) \\ & \sum_{k \in K : f \in \text{WP}_k} \sum_{p \in \mathcal{P}_k^e} n_k^p \leq s_e & e \in E, f \in E \setminus \{e\} \quad & (5.14) \\ & n_k^p \in \mathbb{Z}_+ & k \in K, p \in \mathcal{P}_k \quad & (5.15) \\ & s_e \in \mathbb{Z}_+ & e \in E. \quad & (5.16) \end{aligned}$$

The objective function minimizes the overall protection cost. Constraints (5.13) state that all working traffic must protected. Constraints (5.14) determine the total number of spare channels allocated on each link ensuring that there will be enough capacity to protect non-disjointly routed requests.

A column generation algorithm is also applied to the LP-relaxation of the SPP-M model (SPP-RM) in order to implicitly generate protection paths. For each demand k, the pricing problem that has to be solved consists in finding a path with minimum reduced cost. The reduced cost of protection path p for demand k can be written as:

$$\bar{c}_k^p = -\mu_k - \sum_{e \in p} \sum_{f \in \text{WP}_k} \pi_f^e,$$

where $\mu_k \geq 0$ and $\pi_f^e \leq 0$ are the dual prices of constraints (5.13) and (5.14), respectively. For a given demand k, the term μ_k is constant and therefore the path with the lowest reduced cost for k can be defined as the shortest path in the modified graph $G' = (V, E')$, where $E' = E \backslash \{e : e \in \text{WP}_k\}$ and the link costs are defined by $c'_e = \sum_{f \in \text{WP}_k} -\pi_f^e$.

In [128], the joint optimization of working routing and SBPP in directed networks is considered. The authors presents an LP model solved by column generation. The subproblem consists in computing a pair of paths (working and protection paths) with minimum total cost. Although NP-hard, the subproblem is solved with a dynamic programming based label algorithm in pseudo-polynomial time. Due to the focus of our study on a sequential optimization process, the SPP-RM model can be very easily solved by column generation since the associated pricing problem is a well-known polynomial problem instead of a NP-hard one.

5.3.2.2 FIPP p-cycle model

Link-protecting p-cycles were extended with the goal of providing end-to-end, originating the Failure Independent Path-Protecting (FIPP) p-Cycles [64]. Under FIPP p-cycles, the cyclic protection structures can be shared by a set of working paths for protection as long as the working paths in this set are mutually disjoint or, if they are not, their protection paths are mutually disjoint.

Similarly to the model presented in Section 5.3.1.2, the model for the optimal placement of FIPP p-cycles is based on variables n_p which represent the amount of spare capacity allocated on each link of FIPP p-cycle p. However, there is a important difference between these models: Here, we let variables n_p be possibly associated with the same topological cycle but with different coefficient vectors a^p. This means that there may be several identical cycles providing different levels of protection. For a given request k, coefficient a_k^p is equal to 2 if k is protected for two units by cycle p; 1 if one unit of protection is provided; and 0 otherwise. Note that a request can have two protected units only if it straddles the cycle. However, at most one unit of protection can be provided for concurrent straddling request at the same time.

The FIPP p-cycle model (FIPP-M) is formulated as follows:

$$\min \sum_{p \in \mathcal{P}} \text{COST}_p n_p$$

$$\text{subject to:} \qquad \sum_{p \in \mathcal{P}} a_k^p n_p \geq b_k \quad k \in K \tag{5.17}$$

$$n_p \in \mathbb{Z}_+ \qquad p \in \mathcal{P}. \tag{5.18}$$

Constraints (5.17) are the demand constraints, where $a_k^p n_p$ gives the amount of traffic of demand k protected by p.

The pricing problem is the minimization of the reduced cost subject to the constraints for defining a cycle and for identifying the protected requests. There are two sets of binary variables: The first one contains variables x_e, such that $x_e = 1$ if the cycle traverses link e, and the second one contains variables x_e^k, such that $x_e^k = 1$ if link e is used to protect request k. The pricing problem is then formulated as follows:

$$\min \sum_{e \in E} c_e \, x_e - \sum_{k \in K} \left(\mu_k \sum_{e \in \omega(o_k)} x_e^k \right)$$

subject to:

$$\sum_{e \in \omega(v)} x_e \leq 2 \qquad v \in V \tag{5.19}$$

$$\sum_{e' \in \omega(v): e' \neq e} x_{e'} \geq x_e \qquad v \in V, e \in \omega(v) \tag{5.20}$$

$$\sum_{e \in \omega(V')} x_e \geq 2(x_{e'} + x_{e''} - 1) \qquad V' \subset V, e' \in E(V'), e'' \in E(V \setminus V') \tag{5.21}$$

$$\sum_{e \in \omega(o_k)} x_e^k = \sum_{e \in \omega(d_k)} x_e^k \qquad k \in K \tag{5.22}$$

$$\sum_{e' \in \omega(v): e' \neq e} x_{e'}^k \geq x_e^k \qquad k \in K, v \in V \backslash \{o_k, d_k\}, e \in \omega(v) \tag{5.23}$$

$$\sum_{k \in K: e' \in \text{WP}_k} x_e^k \leq x_e \qquad e \in E; e' \in L \setminus \{e\} \tag{5.24}$$

$$x_e^k = 0 \qquad k \in K, e \in \text{WP}_k \tag{5.25}$$

$$x_e^k \in \{0, 1\} \qquad e \in E, k \in K \tag{5.26}$$

$$x_e \in \{0, 1\} \qquad e \in E \tag{5.27}$$

Constraints (5.19)-(5.21) address the cycle construction. Constraints (5.22)-(5.23) determine the protection route(s) over the cycle for the requests. Next, we have constraints (5.24) that prevent requests from using more protection capacity than it is provided by the cycle under construction, and constraints (5.25) that prevent requests from using a given span in both working and protection paths. The objective function calculates the reduced cost and, as for PC-M model, it is basically composed of two terms: one corresponding to the compound cost of the links used by the cycle, and another one associated with the reward resulting from the requests chosen to be protected.

5.3.3 Implementation details

In order to solve the optimization models described in Section 5.3, some implementation issues must be addressed. First, let us explain how the working paths have been computed. Since working and protection paths must be link disjoint, we must ensure that there exists at least one cycle providing one link-disjoint protection path for each working path. Therefore, for each connection demand, the working path is set to the first shortest path between end nodes, computed by an algorithm for the k shortest paths problem [30], for which there is a node-disjoint alternate path. The same input set of working paths is provided to all models.

The LP relaxation of the restricted master problems of the four models were solved using the CPLEX 10.1.1 solver. The solution approach for the pricing problems varies according to the protection scheme. The pricing problems of the SLP-RM and SPP-RM models were both solved using Dijkstra's algorithm on their respective modified networks in order to find the most profitable protection paths. The pricing problems of PC-RM and FIPP-RM, whose MIP formulations can be found in [127] and [57] respectively, were solved using the CPLEX MIP solver. However, they are not solved to optimality, solver execution is stopped as soon as a solution with negative reduced cost is obtained. Note that this does not hamper the optimality of the solution, instead, it often speeds up the solution process of the master problem. In addition, the solution of the two latter pricing problems require the use of costly subtour elimination constraints, which are only introduced to the set of constraints when they are violated in the incumbent solution.

Initially, all column generation algorithms start with a set of artificial (dummy) columns. An artificial column is a path (in SLP-RM and SPP-RM models) or a cycle (in PC-RM and FIPP-RM models) providing some protection but which is so costly that it will never be part of the optimal solution. The column generation algorithms obtain optimal solutions for the LP relaxation of the protection models, which are not guaranteed to be integer. In order to obtain integer solutions, the MIP models are solved using CPLEX solver with the columns introduced during the column generation process. Although, it is not certain that doing so necessarily leads to the optimal integer solutions, the gap against optimality can be accurately evaluated using the optimal lower bounds from the column generation algorithm.

5.4 Computational results

The four column generation approaches were tested on four real-world networks, all described in Table 5.I. For each network, Table 5.I reports the number of nodes, the number of links, the average node degree, the number of connection requests and the working cost. The network topologies as well as the traffic matrices for the USA and Germany instances can be obtained from [53], and for pan-European Cost239 and Atlanta networks are available in [6] and [90], respectively. The link costs in all networks are considered to be proportional to the geographical distances between cities. Each element of the traffic matrices represents the number of communication channels required between each pair of nodes and it is obtained by dividing the original traffic load by 2.5 Gbit/s. Tests were performed on a AMD 64-bit machine with 16GB of RAM under Linux operating system.

The solutions obtained for all models, using the algorithms discussed in Section 5.3.3, are compared in Tables 5.II and 5.III. The comparison is based on the redundancy of the obtained designs. Redundancy is a measure of architectural efficiency for survivable networks and is measured by the ratio of protection to working cost [38], $\mathcal{R} = \sum_{e \in E} c_e s_e / \sum_{e \in E} c_e w_e$, where w_e and s_e are the number of working and spare channels on

Table 5.I: Real-world networks

Networks	Nodes	Links	Node Degree	Number of Requests	Working Cost
Cost239	11	26	4.70	55	$137,170$
USA	14	21	3.00	91	$5,926,306$
Atlanta	15	22	2.90	105	$151,019$
Germany	17	26	3.06	136	$407,130$

link e, respectively. In other words, the more redundant is the design, the more protection cost it requires. Moreover, we also show the gap between the integer and the optimal linear-relaxed solutions obtained by the column generation algorithms, the number of generated columns and the total computation time (linear relaxation time + integer solution time) for each model. In all cases, we observed very small gaps which proves that our solutions are very close to the optimality.

As expected, path protection schemes are more cost efficient than link protection schemes. For instance, FIPP p-cycles yield between 44.63% and 107.60% extra cost while link-protecting p-cycles require between 55.27% and 113.46%. Also from Tables 5.II and 5.III, it is clear that the denser is the network, the less extra cost is required for protection.

Table 5.II: Results for link protection

Networks	SLP-M				PC-M			
	Redundancy (%)	Gap (%)	Number of columns	Time (s)	Redundancy (%)	Gap (%)	Number of columns	Time (s)
Cost239	54.25	0.26	260	0.13	55.27	0.78	27	0.67
USA	108.28	0.00	49	0.01	113.46	0.00	10	0.19
Atlanta	86.80	0.00	43	0.01	90.22	0.01	15	0.43
Germany	96.53	0.00	71	0.03	111.95	0.00	15	0.49

According to the results, in average, basic p-cycles yield 8.2% more protection cost than classic link protection while the extra protection cost required by FIPP p-cycles in comparison with classic path protection can raise to 20.1%. Although p-cycles and

Table 5.III: Results for path protection

Networks	SPP-M Redundancy (%)	SPP-M Gap (%)	SPP-M Number of columns	SPP-M Time (s)	FIPP-M Redundancy (%)	FIPP-M Gap (%)	FIPP-M Number of columns	FIPP-M Time (s)
Cost239	41.87	0.30	357	3.74	44.63	2.82	260	5376.25
USA	84.22	0.00	153	0.29	95.24	0.10	145	2038.11
Atlanta	82.67	0.00	153	0.13	90.02	0.00	257	1799.99
Germany	79.94	0.01	268	1.05	107.60	0.01	334	2025.14

FIPP p-cycles schemes require some extra cost with respect to the basic link and path protection costs respectively, the difference is rather small on dense networks, such as Cost239, which offers a greater flexibility for choosing cyclic structures. In spite of a slightly higher cost, p-cycle designs remain attractive and offer operational advantages: They present the properties of ring "50 ms" switching as a consequence of pre-cross-connection of paths [38], while restoration times are significantly higher with shared link and protection schemes [34].

Computation times are satisfactorily small for all models, except for FIPP-M. Indeed, this model is not scalable and consumes significantly more time as the network topology and traffic matrix grow. To overcome this issue, we need to devise an efficient approach solution for the pricing problem, which is responsible for 99.99% of time consumption of the linear relaxation.

We performed additional computational tests on a 15-node network family available in [25]. Figure 5.2 shows the redundancy of the designs obtained by the four survivability schemes versus the average node degree for the 15 networks in the test set. Once again, we can see that denser networks take more advantage of spare capacity sharing. We can also observe that in most cases the survivable designs require less protection cost than the working networks: this is the great benefit of spare capacity sharing. Also in these experiments, p-cycles designs are more redundant than the respective classical schemes for all networks. Note that the redundancy does not necessarily decrease as the average node degree increases because working paths can vary in these networks once they are set to be the shortest routes.

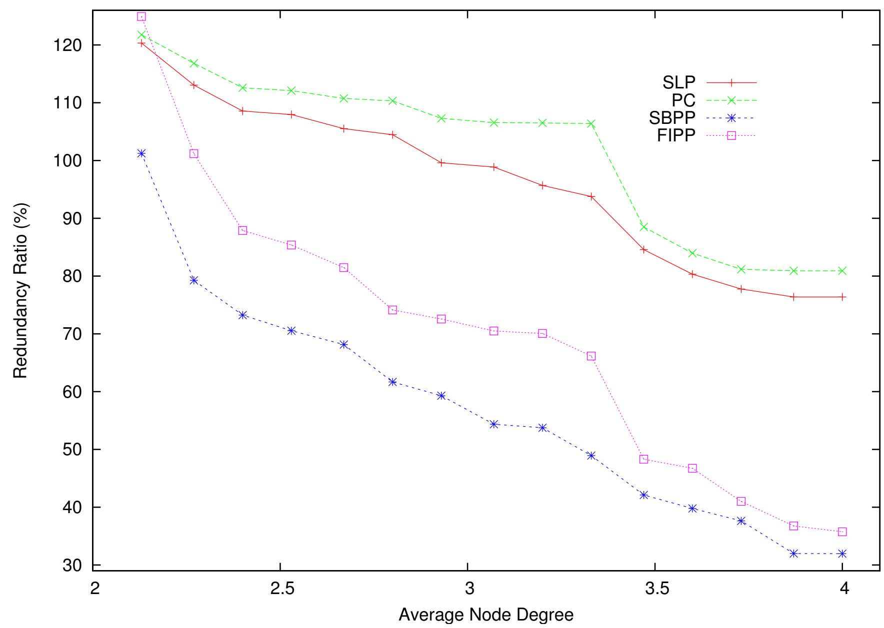


Figure 5.2: Results for 15-node network family.

5.5 Conclusions

We proposed here to investigate the bandwidth protection costs of p-cycles and FIPP p-cycles in comparison with those of classical shared link and path protection, using column generation techniques for solving all four design models without explicit enumeration of cycles and paths. Our goal was to perform a fair and accurate comparison based on optimal or near optimal design solutions.

Results showed that classical protection schemes are more cost efficient than p-cycle schemes, which can be larger than the working costs for some networks. SBPP designs appears as the most cost efficient among all others. More redundant designs are less capacity efficient and, consequently, less costly. Although p-cycle designs are more redundant than classical protection schemes, it is an attractive choice from the operational point of view.

CHAPTER 6

ASYMMETRY ISSUES IN P-CYCLE AND FIPP P-CYCLE PROTECTION SCHEMES

6.1 Chapter presentation

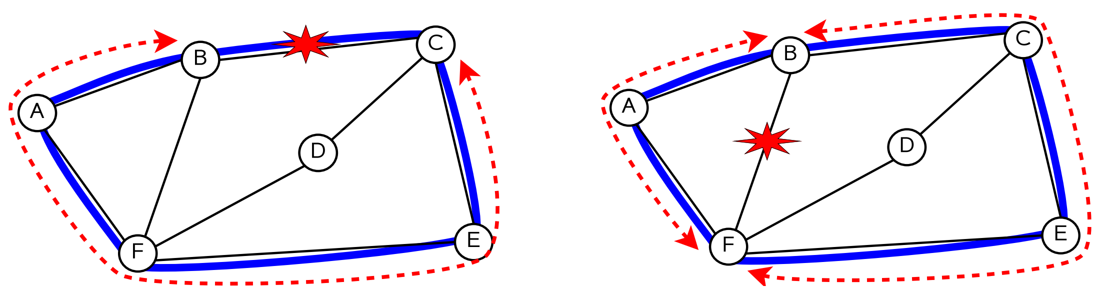
This chapter consists of the homonymous article, co-authored with Brigitte Jaumard and Pierre-Etienne Bougué, which was submitted for publication in *Networks*. A shorter version of this paper with preliminary results was published in the *Proceedings of the International Network Optimization Conference (INOC)* under the title of "Directed vs. undirected p-cycles and FIPP p-cycles" in April 2009.

While it is acknowledged that the carried Internet traffic is very asymmetric, most work on p-cycle and FIPP p-cycle protection schemes have been conducted for symmetric traffic. Thus, the focus of this study is on providing estimations of how much we lose in terms of usage of network resources when using symmetric links in spite of asymmetric traffic. To reach our goal, we propose a unified presentation of directed and undirected models for the p-cycle and FIPP p-cycle protection schemes, using column generation formulations. Results show that the use of undirected models can be very cost ineffective under asymmetric traffic scenarios.

6.2 Introduction

Currently deployed networks, whether metropolitan or backbone, are still inherently symmetric, although the traffic that has to be carried out is asymmetric [58, 131]. The study of [9] mentions that asymmetry ratios on major Internet links may vary from as low as 3:2 to as high as 16:1 at some large Internet exchange points. This means that large portions of network resources may be wasted in one transmission direction under highly asymmetric traffic. Although it entails more complex processes in network configuration and management in order to handle asymmetry, the resulting gain is worth the effort, as investigated in [105]. Not many studies were carried out on the influence of the asymmetric traffic on the underlying optical layer cost, and on the comparison, in terms of cost/bandwidth, between bidirectional and unidirectional line systems [79]. While most provisioning models for operation and protection networks have been developed for both directed and undirected models, the p-cycle protection scheme and its variants have mostly been studied with undirected models, except for few studies [2, 82, 94, 104, 137, 142]. Hence, the objective of this paper is to further investigate directed versus undirected models for p-cycle-based protection schemes and their use for recovering asymmetric traffic.

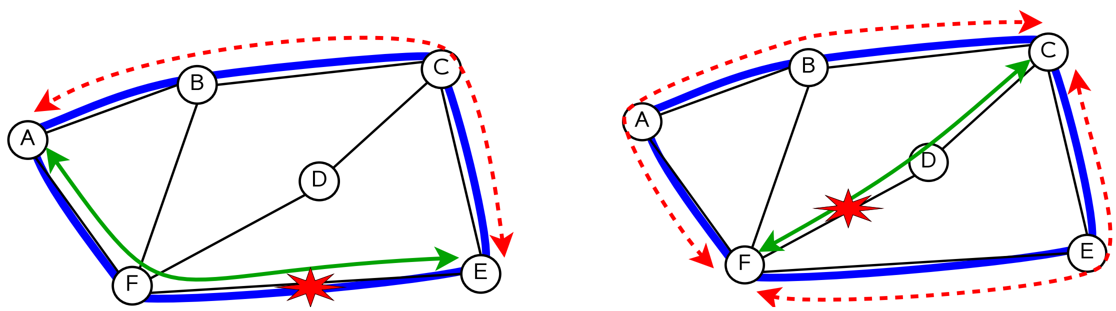
p-Cycles are fully pre-connected cyclic protection structures that protect against straddling link failures, in addition to common ring-like failures [39]. For instance, let us consider the example in Figure 6.1(a) (undirected case). If on-cycle link B-C is perturbed by a failure, the unit-capacity p-cycle offers one alternate path for recovering one unit of working traffic on this link, just like ring protection schemes. Moreover, if a failure occurs on chord link B-F, the unit-capacity p-cycle then offers two alternate paths for recovering two units of traffic on the disturbed link, as illustrated in Figure 6.1(b). What is specially interesting with p-cycles is their pre-cross-connection feature. Indeed, while p-cycles can be viewed as a particular case of the shared link protection (SLP) scheme, they offer much faster recovery delays, comparable to those of ring protection.



(a) On-cycle link failure (b) Chord (straddling) link failure

Figure 6.1: Example of a link-protecting p-cycle. Solid connected lines represent p-cycles and dotted lines with double arrows represent the paths provided for recovering disrupted traffic on failed links.

Link-protecting p-cycles were extended with the goal of providing end-to-end path protection, originating in the failure-independent path-protecting (FIPP) p-cycles [64]. Figures 6.2(a) and 6.2(b) (undirected case again) show how FIPP p-cycles provide protection for end-to-end requests. Under FIPP p-cycles, the cyclic protection structures can be shared by a set of requests for protection as long as their working paths are mutually disjoint or, if they are not, their protection paths over the cycle are mutually disjoint.



(a) On-cycle route failure

(b) Chord (straddling) route failure

Figure 6.2: Example of a path-protecting p-cycle. Solid connected lines represent p-cycles, solid lines with double arrows represent working routes, and dotted lines with double arrows represent the paths provided for recovering disrupted traffic on disturbed requests.

While several different solution methods have been proposed for p-cycles, most works rely on a similar ILP model with the only difference that the set of possible cycles is either generated off-line [31, 31, 39, 42, 71, 118, 138], or generated on-line, often using a column generation algorithm [57, 94, 127]. In [94], the authors propose an exact approach and a polyhedral study of the directed p-cycle placement problem. However, in their study they assume that the spare capacity installed on the p-cycles is fractional, while in this paper we assume this quantity to be integer. In [98], the authors present a collection of mathematical models, following a uniform fashion, for comparing p-cycles and FIPP p-cycles against classical shared link and path protection. The undirected models discussed here were previously presented in [98]. An improved column generation algorithm as well as efficient algorithms for getting integer solutions for undirected FIPP p-cycles are provided in [99]. Note that the focus of the current study is on performing comparisons based on mathematically proved accurate solutions with a scalable solution method.

In order to investigate the behavior of directed systems under asymmetric scenarios, we describe mathematical models for the design of directed and undirected survivable networks using both p-cycle and FIPP p-cycle protection schemes. As exact methods relying on column generation techniques tend to improve the solution scalability, we only focus on those techniques. For integral solutions, we will rely on a heuristic approach which turns out to be effective in producing near optimal solutions (see Section 6.6). Please refer to [14, 75] for a nice background on linear programming and column generation respectively. Moreover, we define two expressions for measuring asymmetry of traffic instances. These metrics allow us to assess the impact on the protection cost caused by the use of directed versus undirected models under different levels of asymmetry.

The paper is organized as follows. Firstly, we present the various models, directed and undirected, for p-cycle and FIPP p-cycle protection, together with models for the pricing problems arising in the column generation algorithms. Thereafter, we present and discuss the experimental results obtained with those models on different network and traffic instances. Results clearly evince that more attention should be given to directed models for p-cycle-based protection schemes.

6.3 Problem definition and assumptions

In the following, we present the problem definition and the assumptions taken into consideration when conducting our research work. First of all, we address the non-joint design problem, i.e., the working traffic is routed in advance using the lowest cost routes in the network so that our focus is on minimizing the overall protection cost for the p-cycles to fully protect the given working traffic. Although the joint optimization of working and protection costs can lead to more resource-efficient designs, one may consider it as an unrealistic assumption since placement of protection capacity is a strategic decision whereas working routing is an operational decision, as discussed in [108]. The following assumptions are additionally taken into account:

– The network topology is biconnected so as to ensure that at least one simple cycle traversing every node exists.
– As for the optimization of directed (FIPP) p-cycles, the network is assumed to be directed, made of unidirectional fiber links. Otherwise, the network is undirected with bidirectional links.
– At most one link can fail at any particular time (single link failure assumption).

In order to simplify the exposition of the different formulations, we will use the same notation for directed and undirected models, except that their definitions may vary depending on whether they are used in a directed or undirected scenario. More particularly, let $G = (V, L)$ be a graph associated with a WDM (wavelength-division multiplexing) network, where V is the set of nodes and L the set of optical (physical) links. Let ℓ refer to an bidirectional link in the undirected models, and to a unidirectional link in the directed models. Note that, when G is a directed graph, each pair of connected nodes is associated with a pair of unidirectional links, one in each direction, denoted by ℓ and $-\ell$. Let us also define $\delta(S), S \subseteq V$, as the cut induced by S, i.e., the set of links incident to a node in S and another node in $V \backslash S$. For a single node $v \in V$, we denote $\delta(v) = \delta(\{v\})$.

The traffic is represented by a set K of connection requests, indexed by k. Bandwidth requirement for each request k is denoted by b_k. In a directed network, the requests are also directed and the traffic is allowed to be asymmetric, i.e., different traffic amounts are required on the opposite directions between a given pair of nodes. In contrast, the requests are assumed to be undirected, and consequently, only symmetric traffic is possible in a undirected scenario.

The overall working traffic on a link ℓ is denoted by $w_\ell = \sum_{k \in K : \text{WP}_k \ni \ell} b_k$, where WP_k is the working route of request k, known beforehand. The total capacity reserved for protection on a link is subject to spare capacity constraints. Let q_ℓ denote the available transport capacity on link ℓ. Then, the spare capacity on link ℓ is given by $q_\ell - w_\ell$.

Furthermore, whether we address a directed or undirected model, we define a set $\mathcal{P}$ of (FIPP) p-cycles, indexed by p, and corresponding variables n_p, which represent the number of copies of unit-capacity p-cycle p or the amount of preconfigured spare capacity of p-cycle p. The cost of a p-cycle p can be defined as $\text{COST}_p = \sum_{\ell \in p} c_\ell$, where c_ℓ is the cost of link ℓ (e.g., the cost of its endpoint interconnection equipment, the distance between its endpoints, etc.).

6.4 p-Cycles

The most relevant difference between directed and undirected p-cycles is that, in the first case, at most one protection unit can be provided by a unit-capacity cycle for each link, as illustrated in Figure 6.4, while with undirected p-cycles, straddling links are provided with two protection units. The extension to FIPP p-cycles is straightforward analogous.

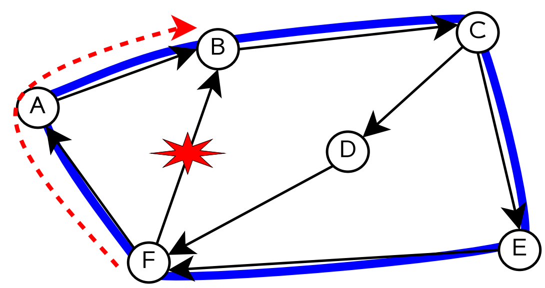


Figure 6.3: Example of an on-cycle link failure in a directed p-cycle.

Before proceeding to the presentation of the ILP model for basic p-cycles, we need to introduce some additional notation, common to both directed and undirected cases. Let coefficients α_ℓ^p be defined as the protection provided by p-cycle $p \in \mathcal{P}$ for link $\ell \in L$. As for undirected p-cycles, coefficients $\alpha_\ell^p \in \{0, 1, 2\}$ such that $\alpha_\ell^p = 2$ if link ℓ straddles cycle p; $\alpha_\ell^p = 1$ if link ℓ is on cycle p; and $\alpha_\ell^p = 0$ otherwise. In the directed case, although the expression of the master problem is identical, the definition of coefficients α_ℓ^p is different: $\alpha_\ell^p \in \{0, 1\}$ such that $\alpha_\ell^p = 1$ if and only if link ℓ has both end nodes on cycle p but is not used by it. Coefficients $\beta_\ell^p \in \{0, 1\}$ define the cycle topology: $\beta_\ell^p = 1$ if and only if link ℓ lies on cycle p in both directed and undirected cases.

The placement of p-cycles in an directed or undirected network can be formulated as follows:

$$\min \quad \sum_{p \in \mathcal{P}} \text{COST}_p \, n_p$$

subject to:

$$\sum_{p \in \mathcal{P}} \alpha_\ell^p \, n_p \geq w_\ell \qquad \forall \ell \in L \tag{6.1}$$

$$\sum_{p \in \mathcal{P}} \beta_\ell^p \, n_p \leq q_\ell - w_\ell \qquad \forall \ell \in L \tag{6.2}$$

$$n_p \in \mathbb{Z}_+ \qquad \forall p \in \mathcal{P}. \tag{6.3}$$

The objective function calculates the overall protection cost. Constraints (6.1) ensure that the working traffic on each link is fully protected. Constraints (6.2) guarantee that the requested bandwidth for establishing working paths and protection cycles does not exceed the link transport capacity. Finally, we have domain constraints (6.3).

Because we aim at solving the model above with on-line generation of the cycles, by using a column generation algorithm, we need to establish the formulation of the pricing problem, i.e., the problem that allows the on-line generation of the most promising cycles. By promising cycle, we mean one with a negative reduced cost. For this, let us introduce three sets of binary variables. The first set contains link variables x_ℓ, $\ell \in L$, such that $x_\ell = 1$ if and only if link ℓ defines one of the links supporting the sought cycle. The second set is made of node variables y_v, $v \in V$, such that $y_v = 1$ if and only if node v belongs to the cycle. Last, the third set is made of protection variables z_ℓ, $\ell \in L$, such that $z_\ell = 1$ if and only if link ℓ is protected by the cycle. Also, let constants $\pi_\ell \geq 0$ and $\lambda_\ell \leq 0$ be the dual prices associated with master problem constraints (6.1) and (6.2) respectively. Note that, in order to provide the pricing problem with dual prices, the linear relaxation of the master problem is solved. In Section 6.6, we explain how integer solutions are obtained.

For ease of understanding, let us distinguish the master problems for directed and undirected p-cycles by naming them PC_ASYM and PC_SYM respectively. The pricing problem for PC_SYM can be formulated as described below.

$$\min \quad \sum_{\ell \in L} (c_\ell + \pi_\ell - \lambda_\ell)\, x_\ell - 2 \sum_{\ell \in L} \pi_\ell z_\ell$$

subject to:

$$\sum_{\ell \in \delta(v)} x_\ell = 2\, y_v \qquad \forall v \in V \tag{6.4}$$

$$\sum_{\ell \in \delta(S)} x_\ell \geq 2\,(y_v + y_{v'} - 1) \qquad \forall S \subset V,\ 3 \leq |S| \leq |V| - 3, v \in S,\ v' \in V \setminus S \tag{6.5}$$

$$z_\ell \leq y_v \qquad \forall v \in V,\ \ell \in \delta(v) \tag{6.6}$$

$$z_\ell \geq y_v + y_{v'} - 1 \qquad \forall v, v' \in V, \ell = \{v, v'\} \in L \tag{6.7}$$

$$y_v \in \{0, 1\} \qquad \forall v \in V \tag{6.8}$$

$$z_\ell, x_\ell \in \{0, 1\} \qquad \forall \ell \in L. \tag{6.9}$$

The expression of objective function, which calculates the reduced cost of cycle variables n_p, is composed of the following terms: The compound costs of the links used by the cycle ($\sum_{\ell \in L} c_\ell\, x_\ell$), the reward gained for protecting requests ($\sum_{\ell \in L} \pi_\ell\, (2z_\ell - x_\ell)$), and the price for using spare capacity ($\sum_{\ell \in L} \lambda_\ell\, x_\ell$). The first two sets of constraints establish conditions to build simple cycles. The degree constraints (6.4) require the degree of each node to be either 0 or 2. Inequalities (6.5) are *connectivity constraints* stating that each cut separating two selected nodes must be crossed twice. Consequently, these constraints ensure that only one cycle is built at a time, as building more than one cycle would entail difficulty to identify the cycle chords. Constraints (6.6) and (6.7) identify the links protected by the cycle. Finally, we have domain constraints. This formulation was first introduced in [130], where this problem was called Quadratic Selective Traveling Salesman Problem, and then, it was used in [127] for solving the p-cycle generation problem in a column generation algorithm for the joint optimization of survivable networks based on p-cycles.

Relations between the variables of the pricing problem and the coefficients of the master problem can be defined as $\alpha_\ell = 2\, z_\ell - x_\ell$ and $\beta_\ell = x_\ell$.

The proposed formulation for the pricing problem in the directed case is slightly different from that for pC_SYM master problem, and it is described below.

$$\min \quad \sum_{\ell \in L} (c_\ell - \lambda_\ell)\, x_\ell - \sum_{\ell \in L} \pi_\ell \, z_\ell$$

subject to:

$$\sum_{\ell \in \delta^+(v)} x_\ell = \sum_{\ell \in \delta^-(v)} x_\ell = y_v \qquad \forall v \in V \tag{6.10}$$

$$\sum_{\ell \in \delta^+(S)} x_\ell \geq y_v + y_{v'} - 1 \qquad \forall S \subset V,\ 3 \leq |S| \leq |V| - 3,\ v \in S,\ v' \in V \setminus S \tag{6.11}$$

$$\sum_{\ell \in \delta^-(S)} x_\ell \geq y_v + y_{v'} - 1 \qquad \forall S \subset V,\ 3 \leq |S| \leq |V| - 3,\ v \in S,\ v \in S,\ v' \in V \setminus S \tag{6.12}$$

$$z_\ell \leq y_v - x_\ell \qquad \forall v \in V,\ \ell \in \delta^+(v) \cup \delta^-(v) \tag{6.13}$$

$$z_\ell \geq y_v + y_{v'} - x_\ell - 1 \qquad \forall v, v' \in V, \ell = (v, v') \in L \tag{6.14}$$

$$y_v \in \{0, 1\} \qquad \forall v \in V \tag{6.15}$$

$$z_\ell, x_\ell \in \{0, 1\} \qquad \forall \ell \in L. \tag{6.16}$$

The change in the objective function is due to the fact that, with directed p-cycles, at most one unit of protection is available for any link, which is simply given by variables z_ℓ. Constraints (6.10) and (6.11-6.12) are the directed counterparts of constraints (6.4) and (6.5) for directed networks. Constraints (6.13) and (6.14) properly identify the links protected by the cycle. Note that a link ℓ cannot be protected by a cycle if it is traversed by this cycle, but the reverse link $-\ell$ can. Relations between the variables of the pricing problem and the coefficients of the master problem can be defined as $\alpha_\ell = z_\ell$ and $\beta_\ell = x_\ell$.

The problem of generating p-cycles was proved NP-hard in [94]. The proof was carried out by reducing the problem to the decision version of the traveling salesman problem.

6.5 FIPP p-cycles

There is an important difference between p-cycle and FIPP p-cycle models concerning the variable definition in the master problem. As for FIPP p-cycles, variables n_p, $p \in \mathcal{P}$, are allowed to be possibly associated with the same topological cycle but with different coefficient vectors α^p. This is motivated by the fact that a FIPP p-cycle may not be able to protect every request with endpoints over it, since requests may be concurrent, i.e., they may not be link disjoint. Therefore, there may be several variables associated with identical cycles but corresponding to different coefficient vectors α.

The master problems of both directed and undirected models (respectively, FIPP_ASYM and FIPP_SYM models) are again the same except for the definition of the coefficient matrix. In the undirected case, the first set of coefficients are defined as follows: $\alpha_k^p \in \{0, 1, 2\}$ such that $\alpha_k^p = 2$ if request k is protected for 2 units by p; $\alpha_k^p = 1$ if request k is protected for 1 unit; and $\alpha_k^p = 0$ otherwise. Indeed, note that a request can be protected for 2 units only if it straddles cycle p. Note that at most one protection unit can be provided for concurrent requests at the same time, even if both of them straddle the cycle (see [57] for details). As for directed FIPP p-cycles, because they can offer at most one unit of protection for a request, coefficients α_k^p are now defined as follows: $\alpha_k^p \in \{0, 1\}$ such that $\alpha_k^p = 1$ if and only if request k is protected by cycle p. Coefficients β_ℓ^p remain the same as for p-cycle models in both directed and undirected cases.

The master problem for directed and undirected FIPP p-cycles is then formulated as follows:

$$\min \quad \sum_{p \in \mathcal{P}} \text{COST}_p n_p$$

subject to:

$$\sum_{p \in \mathcal{P}} \alpha_k^p \, n_p \geq b_k \qquad \forall k \in K \tag{6.17}$$

$$\sum_{p \in \mathcal{P}} \beta_\ell^p \, n_p \leq q_\ell - w_\ell \qquad \forall \ell \in L \tag{6.18}$$

$$n_p \in \mathbb{Z}_+ \qquad \forall p \in \mathcal{P} \tag{6.19}$$

Here, the only difference from PC_ASYM and PC_SYM is in constraints (6.17), since

FIPP p-cycles protect end-to-end requests instead of links. Regarding the pricing problems for FIPP_ASYM and FIPP_SYM, there are some differences between the two models, including the advantage for the directed model to be able to avoid the exponential number of connectivity constraints used to eliminate subtours, whereas containing about twice the number of variables. In both cases, the pricing problem is the minimization of the reduced cost subject to the constraints for defining a cycle and identifying the requests protected by that cycle. There are two sets of binary variables: the first one contains variables x_ℓ, such that $x_\ell = 1$ if and only if the cycle crosses link ℓ, and the second one contains variables x_ℓ^k, such that $x_\ell^k = 1$ if and only if link ℓ is used to protect request k.

The formulation of the pricing problem for FIPP_SYM is described as follows.

$$\min \sum_{\ell \in L} (c_\ell - \lambda_\ell)\, x_\ell - \sum_{k \in K} \left(\pi_k \sum_{\ell \in \delta(s_k)} x_\ell^k \right)$$

subject to:

$$\sum_{\ell \in \delta(v)} x_\ell \leq 2 \qquad \forall v \in V \tag{6.20}$$

$$\sum_{\ell' \in \delta(v) \backslash \ell} x_{\ell'} \geq x_\ell \qquad \forall v \in V, \ell \in \delta(v) \tag{6.21}$$

$$\sum_{\ell \in \delta(S)} x_\ell \geq 2(x_{\ell'} + x_{\ell''} - 1) \qquad \forall S \subset V,\ 3 \leq |S| \leq |V| - 3, \quad \ell' \in L(S), \ell'' \in L(V \setminus S) \tag{6.22}$$

$$\sum_{\ell \in \delta(s_k)} x_\ell^k - \sum_{\ell \in \delta(d_k)} x_\ell^k = 0 \qquad \forall k \in K \tag{6.23}$$

$$\sum_{\ell' \in \delta(v) \backslash \ell} x_{\ell'}^k \geq x_\ell^k \qquad \forall k \in K, v \in V \backslash \{s_k, d_k\}, \ell \in \delta(v) \tag{6.24}$$

$$\sum_{k \in K : \ell' \in \mathrm{WP}_k} x_\ell^k \leq x_\ell \qquad \forall \ell \in L, \ell' \in L \setminus \{\ell\} \tag{6.25}$$

$$x_\ell^k = 0 \qquad \forall k \in K, \ell \in \mathrm{WP}_k \tag{6.26}$$

$$x_\ell^k \in \{0, 1\} \qquad \forall \ell \in L, k \in K \tag{6.27}$$

$$x_\ell \in \{0, 1\} \qquad \forall \ell \in L \tag{6.28}$$

Introduced in [98], this formulation is an improved version of that presented in [57] since it corresponds to a more compact formulation. Regardless of the connectivity constraints which are present in both formulations, the previous formulation has $O(|L||K|^2)$ constraints, whereas this one reduces this number to $O(|L||K|)$ assuming that $|K| > |L|$. Constraints (6.20) and (6.21) ensure that the degree of each node is either 0 or 2, without using node variables. Inequalities (6.22) are a variant of the connectivity constraints (6.5) using only link variables, as this formulation does not contain node variables. Constraints (6.23) and (6.24) determine the path(s) over the cycle for protecting the requests. Finally, inequalities (6.25) prevent requests from using more protection capacity than what is provided by the cycle under construction, and constraints (6.26) prevent requests from using a given link in both working and protection paths. The objective function, which again calculates the reduced cost, is basically composed of two terms: One corresponding to the compound costs of the links used by the cycle, and another one associated with the reward resulting from the requests chosen to be protected.

Relations between the variables of the pricing problem and the coefficients of the master problem can be expressed as $\alpha_k = \sum_{\ell \in \delta(s_k)} x_\ell^k$ and $\beta_\ell = x_\ell$.

Let us recall that the pricing problem for FIPP_ASYM model is allowed to provide columns composed of more than one cycle, therefore there is no subtour elimination constraints in this model. This is made possible by the use of the double-indexed variables together with directed flows in the modeling, which enables a proper identification of protection paths for straddling requests. Note that, while it is possible to get a formulation without the subtour elimination constraints for the other models (see the Appendix), we have favored the forthcoming formulations for their simplicity, overcoming this last difficulty by introducing such constraints in the model only when violated during the solution of the pricing problem. Next, let us present the formulation of the pricing problem for FIPP_ASYM model.

$$\min \sum_{\ell \in L} (c_\ell - \lambda_\ell)\, x_\ell - \sum_{k \in K} \left(\pi_k \sum_{\ell \in \delta^+(s_k)} x_\ell^k \right)$$

subject to:

$$\sum_{\ell \in \delta^+(v)} x_\ell = \sum_{\ell \in \delta^-(v)} x_\ell \leq 1 \qquad \forall v \in V \tag{6.29}$$

$$\sum_{\ell \in \delta^+(v)} x_\ell^k - \sum_{\ell \in \delta^-(v)} x_\ell^k = 0 \qquad \forall k \in K, v \in V \setminus \{s_k, d_k\} \tag{6.30}$$

$$\sum_{\ell \in \delta^-(s_k)} x_\ell^k = 0 \qquad \forall k \in K \tag{6.31}$$

$$\sum_{k \in K : \{\ell', -\ell'\} \cap \text{WP}_k \neq \emptyset} x_\ell^k \leq x_\ell \qquad \forall \ell \in L, \ell' \in L \setminus \{\ell, -\ell\} \tag{6.32}$$

$$x_\ell^k + x_{-\ell}^k = 0 \qquad \forall k \in K, \ell \in \text{WP}_k \tag{6.33}$$

$$x_\ell^k \in \{0, 1\} \quad \forall k \in K, \ell \in L \tag{6.34}$$

$$x_\ell \in \{0, 1\} \quad \forall \ell \in L \tag{6.35}$$

Constraints (6.29) determine that a directed simple cycle is constructed, by ensuring flow circulation at each node and by stating that the outdegree of each node is at most one. Equations (6.30) are flow conservation constraints on the protection paths of each request. Constraints (6.31) play a key role in properly identifying straddling requests. Indeed, they prevent a protection path from terminating at the orign node of a request, otherwise requests with end nodes on different cycles would appear as straddling requests in the objective function. Then, we have constraints (6.32) preventing requests from using more protection capacity than provides the cycle under construction as well as constraints (6.33) preventing requests from using a given span in both working and protection paths. The objective function is expressed similarly to the undirected case. Relations between variables of the pricing problem and coefficients of the master problem can be expressed as $\alpha_k = \sum_{\ell \in \delta^+(s_k)} x_\ell^k$; and $\beta_\ell = x_\ell$.

Proposition 1. *The pricing problem of FIPP p-cycles is NP-hard.*

Proof. The problem of generating basic p-cycles is a special case of FIPP p-cycles where all working paths contain only one link. □

6.6 Computational experiments

All four mathematical models (pC_SYM, pC_ASYM, FIPP_SYM and FIPP_ASYM) were implemented under the assumption of uncapacitated links, as most work in the literature, and solved as proposed using CPLEX 10.1.1 solver. Integer solutions were obtained by providing the solver with the columns generated during the column generation algorithms. Although we cannot claim that this is an exact approach, we obtained optimal or nearly optimal solutions, as shown later. We first describe the traffic and network instances used in the experiments, as well as how we measure the traffic asymmetry. Thereafter, we present the cost discrepancies obtained for the various instances.

A measure of traffic asymmetry needs to be defined in order to assess the effect of asymmetry on the bandwidth cost. Two different ratios are then proposed. The first one attempts at measuring the traffic asymmetry, in terms of sources and destinations, weighted by the amount of bandwidth to be carried out, and is given by $\text{ASYM}_{sd} = 1 - \text{MIN}_{\mathcal{SD}}/\text{MAX}_{\mathcal{SD}}$, where $\text{MIN}_{\mathcal{SD}} = \sum_{sd \in \mathcal{SD}} \min\{b_{sd}, b_{ds}\}$ and $\text{MAX}_{\mathcal{SD}} = \sum_{sd \in \mathcal{SD}} \max\{b_{sd}, b_{ds}\}$. The second one aims at measuring the resulting traffic asymmetry in the carried bandwidth in the network and is given by $\text{ASYM}_{\ell} = 1 - \text{MIN}_{\ell}/\text{MAX}_{\ell}$, where $\text{MIN}_{\ell} = \sum_{\ell \in L} \min\{w_{\ell}, w_{-\ell}\}$ and $\text{MAX}_{\ell} = \sum_{\ell \in L} \max\{w_{\ell}, w_{-\ell}\}$. Once we have defined how to measure the asymmetry of a traffic instance, we developed a traffic generator which takes as input the desired sd-asymmetry percentage and a symmetric traffic instance for a given network. For each symmetric request, the traffic generator randomly calculates reverse request pairs, each one composed of a request with the symmetric demand value and the other one with a random demand value in the opposite direction. The random values are selected in such a manner that the final sum of generated values meet the desired asymmetry ratio. In [79], authors propose a different asymmetry ratio, more precisely, $\text{ASYM}_{sd} = (\text{MAX}_{\mathcal{SD}} - \text{MIN}_{\mathcal{SD}})/(\text{MAX}_{\mathcal{SD}} + \text{MIN}_{\mathcal{SD}})$.

In our experiments, we consider the benchmark network instances listed in Table 6.I. For each network, we provide the number of nodes, the number of undirected links, the average node degree, and the number of undirected requests. Note that, when solving directed models, the values of $|L|$ and $|K|$ are twice those for the directed case. Each request is routed via a shortest path from source to destination. Working routes are carefully assigned so that they are exactly the same in both directions for both directed and undirected models. For each network instance, Table 6.I also shows the traffic and link asymmetry ratios, which are meaningful only for the directed models. The last column says how the asymmetric instances were selected: B means that a benchmark traffic instance was used together with the network instance, and G means that a traffic instance was built with the generator described in the previous paragraph, for the specified asymmetry ratio. The symmetric traffic instances used for generating asymmetric traffic are available in the references provided in the table. It is observed that link asymmetry is often reduced with respect to traffic asymmetry. Most results have been obtained for an average of 10% link asymmetry.

Network	$\|V\|$	$\|L\|$	node degree	$\|K\|$	asymmetry (%) ASYM$_{sd}$	ASYM$_\ell$	traffic instance
COST239 [6]	11	26	4.7	55	20	11	G
POLSKA [90]	12	18	3.0	66	20	7	G
USA [53]	14	21	3.0	91	20	7	G
ATLANTA [90]	15	22	2.9	105	16	13	B
GERMANY [53]	17	26	3.1	136	20	13	G
NEW-YORK [90]	16	49	6.1	240	13	10	B
NORWAY [90]	27	51	3.8	351	46	12	B
EUROPE [53]	28	41	2.9	378	20	6	G
COST266 [90]	37	57	3.1	666	20	3	G
PIORO40 [90]	40	89	4.5	780	20	3	G

Table 6.I: Characteristics of the data sets

Tables 6.II and 6.III show information about the (nearly) optimal protection solutions obtained with directed and undirected models for p-cycles and FIPP p-cycles respectively. For each tested network, the number of cycles, the redundancy ratio (protection over working cost), the optimality gap of the obtained solutions are presented. In addition, the overall number of generated p-cycles (between parenthesis) is also provided. Note that, when comparing directed against undirected models, working costs for the former are assumed to be double those for the undirected case.

It is shown that the redundancy ratios decrease by about 10% with directed p-cycle

Instance	pC_SYM			pC_ASYM		
	# cycles	RR (%)	gap (%)	# cycles	RR (%)	gap (%)
COST239	9 (24)	55.3	0.752	15 (50)	50.3	0.302
POLSKA	8 (12)	81.2	0.002	14 (25)	73.0	0.004
USA	8 (10)	113.5	0.007	15 (20)	102.8⋆	0.000
ATLANTA	8 (13)	90.2	0.008	18 (21)	84.4	0.002
GERMANY	15 (18)	111.9	0.001	16 (29)	100.6⋆	0.000
NEW-YORK	17 (57)	42.4	0.599	62 (17)	40.3	0.052
NORWAY	26 (104)	64.6	0.021	43 (160)	49.4	0.016
EUROPE	16 (33)	109.0⋆	0.000	29 (67)	99.3	0.004
COST266	18 (70)	98.9	0.003	38 (123)	89.0	0.003
PIORO40	29 (171)	61.3	0.010	59 (369)	55.1	0.001

Table 6.II: Results obtained for p-cycles

schemes, meaning that the protection cost is reduced by about 10% when using a directed model instead of an undirected one. Results followed by ⋆ correspond to optimal solutions, i.e, the gap between the lower bound found by the column generation algorithm and these integer solutions is zero. As for the remaining results, the optimality gaps are very small, except for undirected FIPP p-cycle result with COST239 network, whose gap increases to about 3%. We can also observe that solutions obtained with directed models usually have smaller optimality gaps.

Instance	FIPP_SYM			FIPP_ASYM		
	# cycles	RR (%)	gap (%)	# cycles	RR (%)	gap (%)
COST239	11 (260)	44.6	2.818	41 (1109)	40.5	0.499
POLSKA	46 (243)	67.6	0.027	93 (584)	60.6	0.009
USA	63 (315)	95.2	0.109	146 (1003)	86.2	0.001
ATLANTA	82 (257)	90.0⋆	0.000	179 (557)	83.8	0.006
GERMANY	84 (334)	107.6	0.005	170 (969)	96.6	0.003

Table 6.III: Results obtained for FIPP p-cycles

With the purpose of evaluating the reduction in protection costs while increasing traffic asymmetry, additional experiments were performed and the obtained results are illustrated in Figure 6.4. Five different traffic instances with fixed traffic load were generated and with asymmetry ratios ranging from 0% (pure symmetric) to 100% (pure asymmetric, i.e., unidirectional traffic between every pair of source and destination) for

COST239 network. In the graph of Figure 6.4, plotted points correspond to the average ratio of asymmetric over symmetric results obtained using p-cycles or FIPP p-cycles for a given asymmetry percentage. For instance, with an asymmetry ratio of 10%, there is an average gain of almost 5% in the protection cost over the symmetric case when using directed p-cycles. Indeed, the use of asymmetric links can yield an average gain of nearly 45% under pure asymmetric traffic.

It can also be observed that there is an almost linear reduction in protection costs as asymmetry increases, with both p-cycles and FIPP p-cycles. This shows that the use of directed models for protecting asymmetric traffics with p-cycle-based schemes may be of great interest.

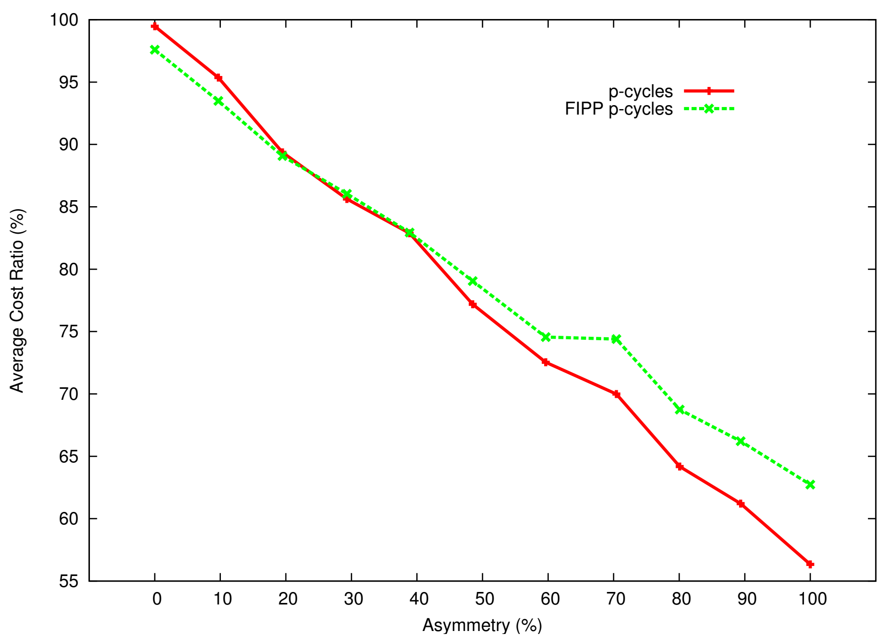


Figure 6.4: Accuracy of the symmetric model for COST239.

6.7 Conclusion

In this paper, we presented mathematical models for the design of survivable networks using directed p-cycles and FIPP p-cycles. The models were solved by means of column generation algorithms and the obtained solutions were compared against those obtained for undirected models. For that, we defined a measure of traffic asymmetry and developed a generator of asymmetric traffic instances.

Our goal was to evaluate the impact on the cost of p-cycle-based networks under asymmetric traffic scenarios. We observed a quite surprising efficiency gain when asymmetric links are used: Reductions of up to 45% in the cost over undirected models. Indeed, the efficiency gain increased linearly as traffic asymmetry augments. In conclusion, the use of asymmetric links is very cost effective under asymmetric traffic scenarios and the difficulty implied by the asymmetry reality in transport networks may be worthwhile.

Some possible improvements in the formulations are being investigated. For instance, the subproblem for directed p-cycles can be formulated without subtour elimination constraints, as for directed FIPP p-cycles. But the benefits of this approach were not observed in preliminary experiences, since it implies more variables and constraints.

6.8 Appendix

It is possible to get rid of the large number of subtour elimination constraints for both directed and undirected p-cycles. We present here those alternate formulations for the pricing problems. As mentioned earlier, although attractive, it turned out in the experiments we did that the resulting formulations were less efficient than the ones presented in the paper, when solving using the so-called "lazy constraints" technique, i.e., when generating those constraints only as needed.

By removing the subtour elimination constraints, the pricing problem can possibly come up with columns composed of more than one cycle, which can mislead the identification of protected straddling links, since a link with end nodes crossed by different cycles is erroneously considered as a straddling link. This issue is overcome by using a directed model with double-indexed variables for representing the protection paths for links, as it is done for FIPP_ASYM. The same directed model can be used to solve the pricing problem for both directed and undirected p-cycles, although different relations are established between variables of the pricing problem and the coefficients of the master problem. Note that, to use a directed model for the undirected case, each bidirectional link ℓ is replaced by two unidirectional links as well as each undirected connection request gives place to a pair of directed connection requests in opposite directions. In this model, links are indexed by $\ell \in L$ as before, and by $e \in L$ when representing working links. Besides variables x_ℓ, the model also contains binary variables x_ℓ^e such that $x_\ell^e = 1$ if and only if protection path of link e goes through link ℓ. The pricing problem for pC_SYM and pC_ASYM models can be further formulated as follows:

$$\min \sum_{\ell \in L} (c_\ell - \lambda_\ell)\, x_\ell - \sum_{e \in L} \left(\pi_e \sum_{\ell \in \delta^+(s_e)} x_\ell^e \right)$$

subject to:

$$\sum_{\ell \in \delta^+(v)} x_\ell = \sum_{\ell \in \delta^-(v)} x_\ell \leq 1 \qquad \forall v \in V \tag{6.36}$$

$$\sum_{\ell \in \delta^+(v)} x_\ell^e - \sum_{\ell \in \delta^-(v)} x_\ell^e = 0 \qquad \forall e \in L, v \in V \setminus \{s_e, d_e\} \tag{6.37}$$

$$\sum_{\ell \in \delta^-(s_e)} x_\ell^e = 0 \qquad \forall e \in L \tag{6.38}$$

$$x_\ell - x_\ell^e \geq 0 \qquad \forall \ell \in L, e \in L \setminus \{\ell, -\ell\} \tag{6.39}$$

$$x_e^e + x_{-e}^e = 0 \qquad \forall e \in L \tag{6.40}$$

$$x_\ell \in \{0, 1\} \qquad \forall \ell \in L \tag{6.41}$$

$$x_\ell^e \in \{0, 1\} \qquad \forall e \in L, \ell \in L \tag{6.42}$$

Constraints (6.36) determine the cyclic structures by ensuring flow circulation on all nodes, and avoid building non-simple cycles. Constraints (6.37) ensure flow conservation at all nodes traversed by the protection path for each link. Constraints (6.38) state that source nodes can be only at the origin of protection paths, which allows us to properly identify protected links since an outgoing flow from the source node of a link correspond to a protection path for that link. Constraints (6.39) state that protection paths can only go through links traversed by the cycle. Finally, constraints (6.39) prevent working links from being used in their protection paths. The objective function is the same as before except for the term corresponding to the amount of protection given to each link, as discussed next.

As for undirected p-cycles, the relations between the variables of the pricing problem and the coefficients of pC_SYM model are defined as follows. Coefficients $\beta_\ell = x_{\ell_1} + x_{\ell_2}$, where ℓ_1 and ℓ_2 are the unidirectional links obtained from bidirectional link ℓ. Analogously, we have coefficients $\alpha_e = \sum_{\ell \in \delta^+(s_{e_1})} x_\ell^{e_1} + \sum_{\ell \in \delta^+(s_{e_2})} x_\ell^{e_2}$. In the directed case, a link in the pricing problem corresponds to exactly the same link in the master problem, which gives us $\alpha_e = \sum_{\ell \in \delta^+(s_e)} x_\ell^e$ and $\beta_\ell = x_\ell$.

CHAPTER 7

A HIERARCHICAL DECOMPOSITION METHOD FOR EFFICIENT COMPUTATION OF PATH-PROTECTING P-CYCLES

7.1 Chapter presentation

This chapter presents an homonymous article, co-authored with Brigitte Jaumard and Thomas Stidsen, which will very shortly be submitted for publication in *Telecommunication Systems*. Herein, we investigate a hierarchical decomposition approach for the design of survivable networks based on failure-independent path-protecting (FIPP) p-cycles. FIPP p-cycles extend link-protection p-cycles by adding the property of providing end-to-end failure independent path switching against either link or node failures. Most existing work on FIPP p-cycles suffer from either a lack of scalability or a lack of information about the quality of their heuristic solutions. In order to overcome those drawbacks, we propose a hierarchical column generation formulation embedding a decomposition and a more compact formulation of the pricing problem. It turns out to be a much more efficient formulation than the previously proposed column generation one. As for the integer solutions, two heuristic methods are proposed. Computational results show that we are able to solve accurately large network and traffic instances by providing lower bounds as well as good optimal integer solutions.

7.2 Introduction

With the growth of Internet services and the enormous bandwidth capability of optical networks brought with DWDM (Dense Wavelength Division Multiplexing) technology, the information society of today relies massively on communication networks demanding more and more high quality service provisioning. In these network architectures, services are constantly exposed to risks of breakdown, either due to human errors or to equipment malfunctions. Therefore, service survivability mechanisms play a crucial role in the deployment of optical networks. Although several types of failures can occur in a network, such as fiber cut, equipment defect or bad operation, the most predominant scenario is single link failures due to fiber cuts by entrepreneurs or by natural disasters [38, 95, 126].

Several schemes have been proposed to achieve survivability by employing protection or restoration in the optical layer. A quite interesting and recent option among these schemes is p-cycle protection [39]. The main concept behind p-cycles is that they recover from on-cycle link failures exactly as BLSR rings, but they protect chord links, also called straddling links, as well. The practical importance of p-cycles is their ability to achieve a good trade-off between capacity efficiency and restoration time since they provide fully pre-cross-connected protection paths over their cyclic structure. This means that, upon a failure detection, no more action, besides switching the affected traffic at the end nodes, is needed for restoration.

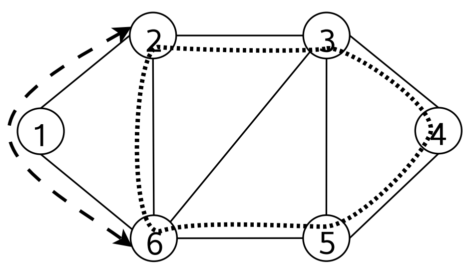
Since p-cycles were introduced in 1998, many studies have been carried out on the topic, including extensions of the p-cycle concept from link protection to node failure recovery [124], path-segment protection [118], and path protection [64], among others. In this study, we are particularly interested in failure-independent path-protecting (FIPP) p-cycles proposed in [64]. FIPP p-cycles extend link-protection p-cycles to allow for end-to-end path protection. Likewise the original concept, the same end-node preplanned protection switching response takes effect.

7.2.1 FIPP p-cycle concept

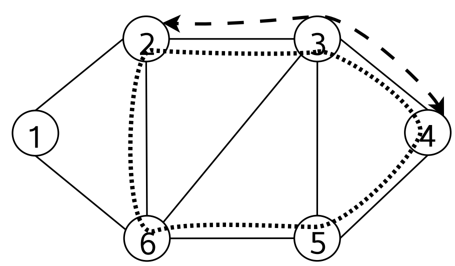
First of all, let us define the concept of working and protection paths. A *working path* is a path used to carry some traffic under normal operation conditions while a *protection path* is a backup path that is only used in case of failure.

The FIPP p-cycle concept is explained using the example illustrated in Figure 7.1. FIPP p-cycles and working paths are represented by dotted and dashed lines respectively. In Figure 7.1(a), path 2-1-6 is a straddling working path since it is link-disjoint from the cycle. Whether link 1-2 or 1-6 fails, protection paths 2-6 and 2-3-4-5-6 over the cycle can be used to restore the traffic on this path. In Figure 7.1(b), a failure on links 2-3 or 3-4 of on-cycle working path 2-3-4 can be recovered by using protection path 2-6-5-4. More complicated relationships between a working path and a FIPP p-cycle can appear as shown in Figure 7.1(c). In this case, called z-relationship, the whole cycle is needed for protecting working path 2-3-6-5 and the protection path used depends on which working link is affected. For example, protection path 2-6-5 can be used to recover from a failure on links 2-3 and 3-6, and protection path 2-3-4-5 protects against a failure on link 5-6.

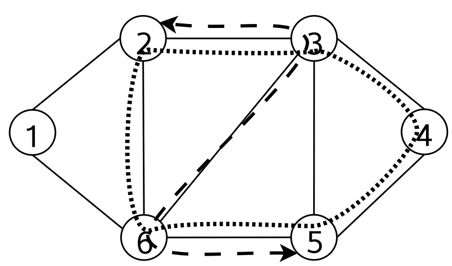
The cyclical protection structure of a FIPP p-cycle can be shared by a set of working paths for protection as long as they are mutually disjoint or, if it is not the case, their corresponding protection paths are mutually disjoint. These criteria have to be met in order to avoid contention for protection resources after a failure. Using our example again, working paths 2-1-6 and 2-3-4 can share the same unit-capacity p-cycle, but working paths 2-3-4 and 2-3-6-5 cannot.



(a) Straddling working path



(b) Fully on-cycle working path



(c) Partially on-cycle working path

Figure 7.1: A FIPP p-cycle example. Working paths are represented by dashed lines and the p-cycle is represented by dotted lines.

7.2.2 Related work

The first work on FIPP p-cycles is documented in [64] and includes an integer linear programming (ILP) model to solve the non-joint design of FIPP p-cycle networks. In this problem, the working paths are routed prior to the placement of FIPP p-cycles. The solution approach consists in first identifying a set of candidate cycles and providing it for the model to determine the best assignment of working paths to cycles with respect to the protection cost. The model, entitled FIPP-SCP, is not scalable since it contains an exponential number of variables and constraints. In their study, the authors assume that only mutually disjoint working paths can share the same p-cycle while one can impose different conditions, see the discussion in the next paragraph.

An alternative approach, called FIPP-DRS, is proposed in [66]. At first, sets of mutually disjoint routes (DRSs) are identified by a heuristic algorithm. Then, a number of candidate cycles is identified for each DRS and all preprocessed data are provided to an ILP model. Both methods in [64, 66] have focused on the idea of mutual disjointness between working routes protected by the same cycle in order to reduce the complexity of the problem. Different assumptions are considered in [57]. Therein, the authors allow non-disjoint paths to be protected by the same cycle but not paths in a z-relationship, see Figure 7.2.2 for an illustration. The authors of [57] proposed a column generation formulation and solution framework for the linear relaxation of the problem, and a heuristic approach to obtain integer solutions. Although this solution approach was shown to be more efficient in obtaining near optimal solutions than the existing methods, the complexity of the pricing problem compromises its scalability. An improved formulation of the pricing problem is then used in [98, 100]. In [98], the authors conduct a systematic comparison between classical shared link and path protection versus p-cycles and FIPP p-cycles. They used a unified column generation formulation and solution scheme to solve all optimization problems. In [100], the authors analyze the impact of the employment of undirected p-cycles for protecting asymmetric traffic on protection costs.

Zhang and Zhong [137] propose a heuristic algorithm for the design of directed FIPP p-cycles. The key idea of their algorithm is to iteratively select the most efficient cycles

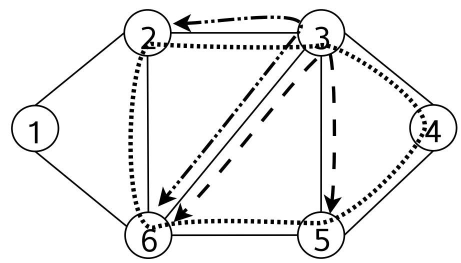


Figure 7.2: Two concurrent working paths (2-3-6 and 6-3-5) sharing a FIPP p-cycle.

from a list of enumerated cycles based on an efficiency metric. This resembles some of the previous approaches proposed for basic p-cycles.

All previously mentioned studies deal with the non-joint optimization FIPP p-cycle design problem, i.e., firstly, the design of the working paths, secondly, the design of the protection scheme. Two further studies, however, turn their attention to joint designs. The first one is due to Ge *et al.* [35]. Therein, a purely heuristic algorithm without any embedded ILP component is used to solve the problem. The method is also based on the enumeration of a set of candidate cycle and DRSs for each candidate cycle. When forming DRSs, multiple working route options are provided for each demand. Then, a routine uses an efficiency metric to select the best assignments of cycles to DRSs.

The second study was carried out by Baloukov *et al.* in [3]. The method resembles an extension of the DRS method proposed in [66]. The overall strategy can be described as follows. For each demand, the N shortest routes are found, composing an eligible route set, where N is an input parameter. Then a modified version of the algorithm from [66] is used to create candidate DRSs by combining selections of route choices that are mutually disjoint into sets. Finally, all these input sets are used for the solution of an ILP formulation.

7.2.3 Contribution

All previous work done on FIPP p-cycles suffer from either their lack of an assessment of the quality of their (heuristic) solutions or a lack of scalability. As shown in [57], some heuristic solutions may be quite far from an optimal solution, sometimes up to 37 %. The objective of this paper is to achieve not necessarily optimal, but provably near optimal solutions, while being able to solve larger data instances. To reach our goal, we propose first an enhancement on the previous column generation formulation for the non-joint design of FIPP p-cycles. It consists of a much more compact formulation of the pricing problem. In addition, we present a new decomposition of the pricing problem which allows us to significantly speed up the run times. Given the large number of columns produced by the column generation algorithm, we also propose two heuristic approaches to obtain integer solutions.

The rest of the paper is organized as follows. In the next section, we formally define the FIPP p-cycle optimization problem, and then a mathematical formulation is presented. In Section 4.7.2, the pricing problem is formally defined. Also, we present a new enhanced formulation and a decomposition of the pricing problem. The column generation algorithm is described in detail in Section 7.5 and the algorithms proposed to produce integer solutions are presented in Section 7.6. Finally, computational experiments are presented in Section 7.7, followed by the conclusions of the work.

7.3 Design of FIPP p-cycle networks

In the following, the problem of designing survivable networks using FIPP p-cycles is formally described. For this, some important assumptions and definitions are discussed and presented in Section 7.3.1, and then a mathematical formulation of the problem is provided in Section 7.3.2.

7.3.1 Assumptions and definitions

We consider an optical network represented by an undirected graph $G = (V, L)$ where V is the set of nodes and L is the set of links, which represent physical entities that collect all channels between neighbor nodes. For instance, a link can represent a set of cables co-routed in the same duct and each cable may contain multiple fibers. There is a linear cost c_ℓ for using one unit of spare capacity of each link $\ell \in L$. Link costs can represent information such as the cost of the interconnection equipment at endpoints, link length, etc. Furthermore, we are given a static set of connection requests, each of them associated with a distinct pair of nodes. For each connection request $k \in K$, we are given its required bandwidth b_k (number of optical channels) and its working route WP_k between its end nodes, o_k and d_k. Let $\mathcal{P}$ denote the set of all potential candidate cycles in the network. The cost of cycle $p \in \mathcal{P}$ is given by $\text{COST}_p = \sum_{\ell \in p} c_\ell$.

The following assumptions are taken into consideration in this study:

Assumption 1. The network is composed of bidirectional (undirected) links, i.e., each topological link consists of a pair of fiber links, one in each direction. We also assume that the traffic is symmetric since the connection requests are undirected.

Assumption 2. Although FIPP p-cycles can provide resilience against node failures, we only address single link failures, which are the predominant failure scenario in optical networks [38]. Thus, whenever we refer to route disjointness hereinafter, we mean link disjointness, except where otherwise stated.

Assumption 3. The working traffic of each connection request is assumed to take a single route. However, we assume that the traffic can be split into integer parts (integer numbers of channels) during protection. For instance, the working traffic of a given connection can be restored on one or more protection routes, possibly over different p-cycles, each one carrying an integer fraction of the failed traffic.

Assumption 4. The working traffic is sent through the lowest cost routes, computed *a priori*, therefore our focus is on determining the cycles needed to protect those working paths.

The FIPP p-cycle design problem can now be defined as follows:

Problem 1 (FIPP p-cycle design problem)**.** Given the above definitions and assumptions, determine a set of FIPP p-cycles with associated protection capacity so as to fully protect all connection requests in K while minimizing the overall protection cost, given by $\sum_{p \in \mathcal{P}} \text{COST}_p s_p$, where s_p is the number of unit copies of cycle p.

7.3.2 An ILP formulation

We now present an ILP formulation for the FIPP p-cycle design problem under the above assumptions. For each cycle $p \in \mathcal{P}$, let us define set $\mathcal{C}_p$ as the set of all its cycle configurations.

Definition 1. A *cycle configuration* corresponds to an association of a unit-capacity p-cycle and a subset of connection requests for which protection is provided. In particular, configuration $C \in \mathcal{C}_p$ is represented by vector α^C of length $|K|$, where coefficients $\alpha_k^C \in \{0,1,2\}$, $k \in K$, define the level of protection (the number of protection paths) provided by cycle p for connection k.

Let us recall that a connection can have two protection paths over a given cycle only if it fully straddles that cycle. In this study, we assume that a feasible cycle configuration meets the following requirements:

i) It corresponds to a simple and unit-capacity cycle;

ii) Only mutually disjoint connections are protected.

The formulation is based on the decision variables n_C, $C \in \mathcal{C}_p, p \in \mathcal{P}$, representing the number of unit copies of p-cycle p reserved for configuration C. The FIPP p-cycle design problem can be formulated as follows:

$$\text{minimize} \quad \sum_{p \in \mathcal{P}} \left(\text{COST}_p \sum_{C \in \mathcal{C}_p} n_C \right) \tag{7.1}$$

subject to:

$$\sum_{p \in \mathcal{P}} \sum_{C \in \mathcal{C}_p} \alpha_k^C n_C \geq b_k \quad \forall k \in K \tag{7.2}$$

$$n_C \in \mathbb{Z}_+ \qquad \forall C \in \mathcal{C}_p, p \in \mathcal{P}. \tag{7.3}$$

The objective function minimizes the cost of the overall capacity used for protection. The demand constraints (7.2) ensure that enough capacity to protect each connection is allocated over all cycle configurations. Finally, we have constraints (7.3) defining the integer domain of the variables.

The main drawback of this model is that the number of possible cycles as well as cycle configurations grows exponentially with the network and traffic sizes. It clearly makes impracticable the optimization of this model through explicit enumeration of all cycle configurations. Fortunately, cycle configurations can be iteratively computed by means of a column generation algorithm so that, as a result, only a small fraction of cycles and configurations will be generated at the end of the process. In order to apply column generation to implicitly consider cycle configurations, the model above is relaxed by removing the integrality constraints, i.e, $n_C \in \mathbb{R}_+, C \in \mathcal{C}_p, p \in \mathcal{P}$. The resulting model is the so-called master problem. When only some columns are considered in the master problem, this is called the restricted master problem (RMP). Columns are further generated based on their reduced costs, which are derived from the dual variables of RMP. The problem of finding negative reduced cost columns to be added to RMP is discussed in Section 4.7.2, and the proposed column generation algorithm is described in detail in Section 7.5. For background information about linear programming techniques, we refer the reader to [14, 23].

7.4 Generating columns

The column generation algorithm iteratively requires the solution of the pricing problem. Assuming that the optimization problem under concern is a minimization one, by definition, the pricing problem corresponds to the problem of finding a column with minimum reduced cost, i.e., to the problem of generating an *augmenting* column that allows decreasing the current value of the objective function. Here, a cycle configuration with minimum reduced cost can be viewed as the one making the best compromise between protection cost and provision of protection for the connections.

Let $\lambda_k \geq 0$, $k \in K$, be the dual variables associated with constraints (7.2). The reduced cost of a cycle configuration C, $C \in \mathcal{C}_p$, given by expression (7.4), is composed of two terms: the cost of cycle p and the revenue obtained for protecting connections.

$$\overline{\text{COST}}_C = \text{COST}_p - \sum_{k \in K} \lambda_k \alpha_k^C. \tag{7.4}$$

As mentioned, the pricing problem corresponds to the problem of finding a cycle configuration with a negative reduced cost, but it can be further decomposed thereby restraining the search to a set of mutually disjoint requests for a given cycle. At first, let us present a formal definition as well as a mathematical formulation for the original pricing problem. Then, we will explain how it can be decomposed and present a formulation for the resulting pricing problem.

The pricing problem, hereinafter called the cycle configuration problem (CCP), can be defined as follows:

Problem 2 (Cycle configuration problem)**.** Let us consider an undirected graph $G = (V, L)$ with link costs $c : L \mapsto \mathcal{R}_+$. Also, let us denote by K the set of requests with associated revenues $\lambda : K \mapsto \mathcal{R}_+$ and working routes WP_k between end nodes o_k and d_k for each request $k \in K$. The cycle configuration problem asks for a cycle configuration of minimum total cost, where the cost of configuration C associated with cycle p is defined by expression (7.4).

As discussed in Section 6.2, different assumptions can be considered with respect to the connections allowed to be protected by a given FIPP p-cycle. More precisely, the connections can be in a "z"-relationship with its protecting cycle [64, 66] or not [57, 100]. Also, only mutually disjoint connections can be protected by the same cycle [64, 66] or not [57, 100]. In this work, we take into account the same assumptions as in [64, 66].

The following notation is introduced in order to present a formulation for CCP. Let us define binary variables x_ℓ, $\ell \in L$, and y_v, $v \in V$, as follows: $x_\ell = 1$ if and only if link ℓ belongs to the cycle; $y_v = 1$ if and only if node v is traversed by the cycle. In addition, let us define binary variables z_k and w_k, $k \in K$, as follows: $z_k = 1$ if and only if connection k is protected; $w_k = 1$ if and only if connection k is protected and straddles the cycle (k is a chord). Let us also define $\delta(S), S \subset V$, as the cut induced by S, i.e, the set of links incident to a node in S and another node in $V \backslash S$. For a single node $v \in V$, we denote $\delta(v) = \delta(\{v\})$. Additionally, let $L(S), S \subset V$, be the set of links induced by S, i.e., the set of links whose both adjacent nodes belong to S. The problem of generating cycle configurations can be formulated as follows:

$$\text{minimize} \quad \sum_{\ell \in L} c_\ell x_\ell - \sum_{k \in K} \lambda_k (z_k + w_k) \tag{7.5}$$

subject to:

$$\sum_{\ell \in \delta(v)} x_\ell = 2y_v \qquad \forall v \in V \tag{7.6}$$

$$\sum_{\ell \in \delta(S)} x_\ell \geq 2(y_i + y_j - 1) \quad \forall i \in S, j \in V \backslash S, S \subset V, \quad 3 \leq |S| \leq |V| - 3 \tag{7.7}$$

$$z_k \leq y_v \qquad \forall k \in K, v \in \{o_k, d_k\} \tag{7.8}$$

$$2w_k + x_\ell - z_k \leq 1 \qquad \forall k \in K, \ell \in \text{WP}_k \tag{7.9}$$

$$\sum_{k \in K \,:\, \text{WP}_k \ni \ell} z_k \leq 1 \qquad \forall \ell \in L \tag{7.10}$$

$$z_k, w_k \in \{0, 1\} \qquad \forall k \in K \tag{7.11}$$

$$x_\ell \in \{0, 1\} \qquad \forall \ell \in L \tag{7.12}$$

$$y_v \in \{0, 1\} \qquad \forall v \in V \tag{7.13}$$

The objective function (7.5) calculates the reduced cost. Degree constraints (7.6) require the degree of each node to be either 0 or 2. Inequalities (7.7) are connectivity constraints forcing each cut separating two visited nodes to be crossed at least twice. Constraints (7.8) ensure that each protected connection has both end nodes crossed by the cycle. Constraints (7.9) determine whether a protected connection is a chord of the cycle or not. Inequalities (7.10) are capacity constraints ensuring that concurrent connections are not simultaneously protected. Therefore, they prevent connections from using more than one unit of protection capacity. Finally, domain constraints (7.11), (7.12), and (7.13) ensure binary values for all variables.

This formulation can be considered as an improvement with respect to the one proposed in [100] since it has a much smaller number of variables and constraints. In the previous formulation, there are variables representing the protection paths taken over the cycle by the connections so as to avoid contention for capacity by non-disjointly routed requests. This is not the case here because we do not allow non-disjoint requests to share the same unit p-cycle, which is ensured by constraints (7.10). In Section 7.7, we will show that the new formulation is indeed more efficient than the previous one.

A shortcoming of the above formulation, which is the major contributor to its high complexity, lies in the exponential number of constraints (7.7) used to avoid subtours (or, in other words, multiple cycles). Those subtour elimination constraints are needed because generating more than one cycle at a time would make the identification of the straddling links very difficult. However, these constraints can be added to the model only when needed using a branch-and-cut algorithm, as explained in Section 7.5.

The cycle configuration problem was proved NP-hard in [101]. Note, however, that the pricing problem does not need to be solved exactly at each iteration in order to obtain an improved solution, and hence heuristic generation of improving columns may be applied. Ultimately optimal solution of the pricing problem is still required to prove optimality of the column generation algorithm, nevertheless.

With this in mind, an interesting finding allows us to efficiently compute heuristic solutions for CCP. The key idea is to decompose the problem into two subproblems: find a cycle and then identify the most profitable configuration for that cycle. Indeed, if the cycle associated with the best configuration to be added to RMP is known, then one only needs to find the set of mutually disjoint requests that can be protected by that cycle and maximizes the revenues from the dual prices. The corresponding subproblem is called the cycle packing problem. Indeed, a set of candidate cycles is known at each iteration of the column generation algorithm: Those associated with the previously generated configurations. The main advantage of this decomposition is that the complexity of this subproblem is greatly reduced in comparison with the aggregate pricing problem. Thus, the resulting column generating algorithm turns out to be less time consuming and more scalable. Note that an approximate solution for the Problem 2 is obtained as a result of this decomposition unless the optimal cycle for the given dual prices is known.

The cycle packing problem (CPP) can be formally defined as follows:

Problem 3 (Cycle packing problem)**.** For a given FIPP p-cycle p, let us define set $K_p = \{k \in K$ such that o_k and d_k are crossed by $p\}$. Each connection request $k \in K_p$ is associated with a nonnegative revenue q_k defined as follows: $q_k = \lambda_k$ if connection k is partially or fully on cycle p, otherwise $q_k = 2\lambda_k$. CPP asks for the subset of mutually disjoint requests in K_p, called a *cycle packing*, that maximizes the total revenue.

We strongly believe that the cycle packing problem is NP-hard, given its similarity to the maximum weight independent set problem (MWISP) [10].

In the following, we present a mathematical formulation for CPP. It also uses binary variables z_k, $k \in K_p$, such that $z_k = 1$ if and only if request k is protected by cycle p.

$$\text{maximize} \quad \sum_{k \in K_p} q_k z_k \tag{7.14}$$

subject to:

$$\sum_{k \in K_p :\ \text{WP}_k \ni \ell} z_k \leq 1 \qquad \forall \ell \in L \tag{7.15}$$

$$z_k \in \{0, 1\} \qquad \forall k \in K_p \tag{7.16}$$

The objective function (7.14) calculates the total revenue and constraints (7.15) ensure that only mutually disjoint requests are protected. Remark that this formulation relates to the classical formulation for MWISP in a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ in which there are constraints $z_i + z_j \leq 1, \forall \{i, j\} \in \mathcal{E}$, stating that two adjacent nodes in the graph cannot be part of an independent set. However, CPP formulation is certainly tighter than that one since it gathers information from the working paths.

Let us suppose a graph on which each edge $\{i, j\}$ is present if requests i and j share the same link in their working paths. Therefore, a set of requests sharing the same link forms a clique in this graph of request clashes. These cliques correspond to constraints (7.15) in the CPP formulation, possibly involving more than two variables. Such valid inequalities are referred to as *clique constraints* for the MWISP. Figure 7.4 illustrates a CPP instance as well as its corresponding graph of request clashes. The set of constraints (7.15) obtained from the links in the example are:

$$\{2,4\} : z_2 + z_5 \leq 1$$
$$\{3,4\} : z_3 + z_5 \leq 1$$
$$\{4,5\} : z_1 + z_2 + z_3 + z_4 \leq 1$$
$$\{5,6\} : z_1 + z_2 \leq 1 \qquad \text{(redundant)}$$
$$\{5,7\} : z_4 \leq 1 \qquad \text{(redundant)}$$

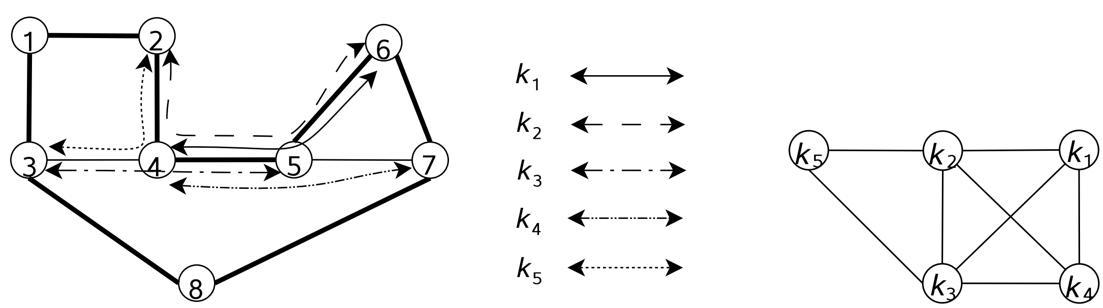


Figure 7.3: A CPP instance (on the left) and its corresponding graph of request clashes (on the right).

Constraints (7.15) tend to involve more variables (represented by large cliques) as more requests compete for the same resources in the network. Note that constraint $z_1 + z_2 + z_3 + z_4 \leq 1$ is associated to a maximal clique in the graph. Clique constraints associated with maximal cliques were proved to be facets of the MWISP polyhedron in [92].

7.5 Column generation algorithm

The idea behind column generation technique is to only introduce variables when needed, i.e., when their reduced cost is negative. The method relies on a decomposition of the initial linear program into a master problem and a pricing problem. The master problem corresponds to a linear program subject to a first set of explicit constraints and a second set of implicit constraints expressed throughout properties of the coefficients of the constraint matrix. The pricing problem consists in the optimization of the so-called reduced cost subject to the set of implicit constraints: It either identifies favorable columns to be added to the master problem or indicates that no such column exists.

The general framework of the proposed column generation algorithm is presented in Figure 7.4. Initially, the algorithm starts with a set of artificial (dummy) columns, one for each request. An artificial column corresponds to a cycle configuration providing protection for only one request, which is so costly that it will never be part of the optimal solution. Then, the restricted master problem containing the initial set of variables is solved and the resulting dual variables are used to guide the search for a cycle configuration with negative reduced cost. This process continues until no improving cycle configuration is found.

A typical iteration of our column generation works in the following fashion. Firstly, each known cycle is considered, from the last to the first generated one, and its corresponding cycle packing problem is solved. As soon as the solution of a given CPP produces an improving column, this is provided to the restricted master problem and a new iteration begins. Contrariwise, if no improving column can be found with the known cycles, we cannot yet claim optimality of the master problem and CCP ought to be solved. If the solution of CCP does not produce an improving column either, the optimal solution for the restricted master problem is also optimal for the original master problem and the algorithm terminates. Otherwise, the column with negative reduced cost is provided for the RMP and the algorithm iterates again.

Solving CPP is the bottleneck of the column generation algorithm. The solutions for CPP are obtained by solving the proposed formulation which unfortunately contains the

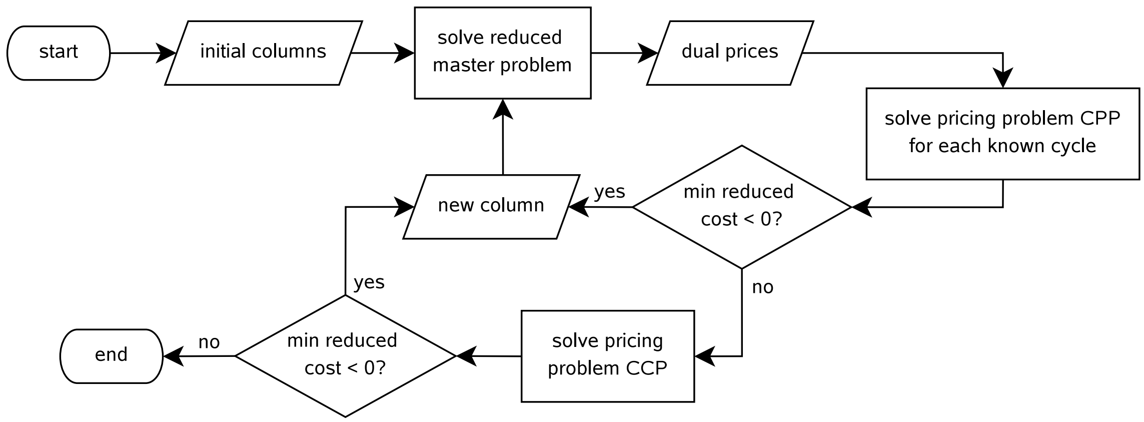


Figure 7.4: Flowchart of the column generation algorithm.

costly connectivity constraints. In order to avoid this pitfall, the model is solved with a branch-and-cut solver using a cutting plane scheme so as the connectivity constraints are only introduced to the model when violated in the incumbent solution. The most violated constraints are identified by the exact separation algorithm proposed in [32], which is based on the computation of maximum flows in the network. Although, after introduced, some of these constraints could be eliminated, they are all kept in the model as the process continues. This offers the advantage that, in practice, the solution of the pricing problem is very often feasible at each iteration, even if only a small number of these constraints have been explicitly introduced. Let us recall that CPP does not need to be solved exactly as long as we are able to find a cycle configuration with negative reduced cost for improving the solution of RMP. Hence, the solution of CPP is stopped as soon as a column with negative reduced cost is obtained. This approach does not hamper the optimality of the master problem solution, instead, it often speeds up the solution process.

All CPPs are solved using the proposed formulation within a branch-and-cut solver.

7.6 Solving the integer problem

The column generation algorithm obtains an optimal solution for the LP relaxation of the FIPP p-cycle optimization problem, which is not guaranteed to be integer, and consequently, is not an optimal solution for the problem of interest. Two alternatives are then proposed here to obtain integer solutions based on the optimal relaxed solution found by the column generation algorithm. The first approach, presented in Section 7.6.1, is a rounding-based algorithm and relies on the iterative solution of LP problems. The second alternative is presented in Section 7.6.2 and is based on the iterative solution of small IP problems following a round-robin fashion.

7.6.1 Rounding-based algorithm

Because integer programming problems are, in general, much more difficult to solve than linear programming problems, a simple and common approach to solve an IP problem is to assume that all variables are permitted to take a real value, to solve the resulting LP problem, by applying the simplex method for example, and then to round off the fractional values of the optimal solution to the nearest integers. However, one pitfall with this approach is that the optimal LP solution is not necessarily feasible after it is rounded. In order to overcome this issue, instead of rounding off all fractional variables at once, an alternative approach is to iteratively rounding off one or few fractional values and to resolve the LP problem with the fixed rounded variables until an integer solution is obtained.

The proposed rounding algorithm is given below.

Algorithm 7.6.1 Rounding-based algorithm

Input: Optimal linear solution n^{LP}

Output: Integer solution n^{IP}

$n^{IP} \leftarrow n^{LP}$
for all non-fixed variable n_C^{IP} **do**
    Initialize coefficient β_C
end for
while $\exists n_C^{IP} \notin \mathbb{Z}_+$ **do**
    Select non-fixed variable n_C^{IP} with best coefficient β_C
    $n_C^{IP} \leftarrow \text{ROUND}(n_C^{IP})$
    Solve modified LP model using the column generation algorithm
    Get current solution n^{IP}
    for all non-fixed variable n_C^{IP} **do**
        Update coefficient β_C
    end for
end while

The algorithm starts with the optimal LP solution found by the column generation algorithm. If the optimum values in that solution are all integers, an integer solution has been found and there is no need to proceed. On the other hand, if any variable has a fractional value, one of them is chosen according to a given criterion and is rounded to its closest integer. Different criteria may be used to choose the variables to be rounded, e.g., its value (number of unit copies) in the linear solution, the protection cost or the amount of protection provided by the corresponding cycle. More precisely, a coefficient β_C is assigned to each non-fixed variable n_C, so that the variable with the largest coefficient at each iteration is chosen. Then, the modified LP model with the just-fixed variable is reoptimized using the column generation algorithm and the coefficients are re-evaluated according to the new current solution. This process continues until there is no variable with fractional value.

7.6.2 Round-robin algorithm

The second approach is inspired by the straightforward idea of that smaller problems tend to be easier to solve. With this is mind, the proposed algorithm is based on a decomposition of a large IP problem into small ones, which are solved following a round-robin fashion. Starting from a empty model, subsets of columns are progressively added to the model, while columns appraised as unpromising are removed, keeping the size of the model always approachable.

A more detailed description of the proposed algorithm is presented by the pseudo-code of Algorithm 7.6.2.

Algorithm 7.6.2 Round-robin algorithm

Input: Sorted list $\mathcal{C}$ of columns; number of columns θ

Output: Integer solution n^{IP}

$n^{IP} \leftarrow \mathbf{0}$
$\mathcal{C}' \leftarrow$
while solution n^{IP} has been improved **do**
    $counter \leftarrow 0$
    while $counter < |\mathcal{C}|$ **do**
        Remove all columns associated with variables equal to zero from $\mathcal{C}'$
        Add next θ columns in $\mathcal{C} \backslash \mathcal{C}'$ to $\mathcal{C}'$
        Solve IP model with columns in $\mathcal{C}'$
        Get current solution n^{IP}
        $counter \leftarrow counter + \theta$
    end while
end while

Initially, the list $\mathcal{C}$ of all columns (configurations) is sorted in such a order that most promising columns appear first in the list according to a given criterion, e.g. most used columns in the relaxed solution. The empty IP model is then initialized with the first θ columns in $\mathcal{C}$, where θ is the number of columns to be inserted at each iteration. Once the first model is solved, columns associated with variables whose value is zero in the final solution are removed from the model and the following θ columns in $\mathcal{C}$, not in the model, are inserted. The new current model is solved and another iteration begins. This process continues until no improvement can be done to the current solution after all columns have been considered.

The list of columns is implemented as a circular structure so that, when the end of the list is attained, the head of the list is reconsidered. Note that, only at the first iteration, the columns to be inserted need to be carefully selected in order to obtain a feasible model.

7.7 Computational results

In this section, we evaluate the efficiency of our solution approach using a set of 16 benchmark problem instances whose details are provided in Table 7.I. For each network, the number of nodes, the number of links, average node degree, and the number of connection requests are provided. When only asymmetric traffic was available for a given instance, we considered the maximum amount of traffic between each pair of nodes in order to obtain a symmetric traffic matrix. The working route for each request was obtained by using Dijkstra algorithm to find the lowest cost route. All algorithms were implemented in C++ programming language using Concert Technology library and version 10.1 of CPLEX solver. The computational experiments were performed on a AMD 64-bit machine with 16GB of RAM.

7.7.1 Column generation algorithm

The first performed experiments evaluate the proposed column generation algorithm with the new formulation for CCP but without the embedded pricing problem decomposition, i.e, the CPP component was not included in the algorithm. Table 7.II provides information about the performance of the solution approach. In more details, we provide the number of columns (cycle configurations) and the number of distinct cycles generated during the column generation process. We also provide the total running time in seconds as well as the percentage of time required for the solution of the master and pricing problems. In the worst case, the column generation algorithm took more than one

Instance	Nodes	Links	Avg. Node Degree	Requests
9N17S [41]	9	17	3.8	36
DFN-BWIN [90]	10	45	9.0	45
COST239 [6]	11	26	4.7	55
POLSKA [90]	12	18	3.0	66
USA [53]	14	21	3.0	91
ATLANTA [90]	15	22	2.9	210
GERMANY [53]	17	26	3.1	136
NEWYORK [90]	16	49	6.1	240
EON-19 [41]	19	37	3.9	171
TA1 [90]	24	55	4.6	163
NORWAY [90]	27	51	3.8	351
BRAZIL [87]	27	70	5.2	351
EON-28 [53]	28	41	2.9	378
BT [41]	30	59	3.9	435
CSELT [41]	30	56	3.7	435
COST266 [90]	37	57	3.1	666

Table 7.I: Characteristics of the problem instances

day to obtain the optimal solution for the linear relaxation of the FIPP p-cycle design problem. From the information provided in Table 7.II, we can also notice that the solution of CPP is very time consuming, responding for more than 99% of the total running time in most cases.

In order to appraise the benefits of the pricing problem decomposition, the performance of the resulting column generation algorithm can be evaluated in Table 7.III. For each tested network, the table shows, besides the number of columns and cycles generated, the running time and the percentage of the time required for solving each component of the algorithm (master problem, CCP, and CPP). We can see now that the total running time is significantly reduced in comparison with the algorithm without pricing decomposition. Indeed, the pricing decomposition yields a reduction of up to 90% in the running time. The decomposition also produces a more even distribution of consumed time among the components. Because the solutions obtained for CCP tend to be suboptimal with respect to CPP, a larger number of columns needs to be generated in order to reach optimality.

Table 7.II: Performance of the column generation algorithm without pricing decomposition

Instances	cost	RR	# columns	time (s)	master time	CCP time
9N17S	2,900.00	49.15%	78	4.68	0.43%	99.57%
DFN-BWIN	178,550.00	52.40%	19	0.58	0.01%	99.99%
COST239	60,263.91	43.93%	243	38.44	0.31%	99.69%
POLSKA	3,398,548.00	72.42%	165	12.86	0.62%	99.38%
USA	5,874,273.44	99.12%	253	49.32	0.43%	99.57%
ATLANTA	135,951.00	90.02%	232	33.46	0.36%	99.64%
GERMANY	446,372.50	109.64%	331	86.19	0.28%	99.72%
NEWYORK	485.93	33.82%	598	347.26	0.24%	99.76%
EON-19	89,489.39	98.05%	930	371.26	0.57%	99.43%
TA1	5,576,871.50	84.78%	868	402.52	0.54%	99.46%
NORWAY	5,504.75	51.01%	2,798	14,051.83	0.25%	99.75%
BRAZIL	1,905,574.40	70.84%	2,990	18,043.30	0.23%	99.77%
EON-28	1,908,707.50	106.24%	2,166	6,057.35	0.23%	99.77%
BT	2,752.69	42.72%	6,155	84,705.04	0.54%	99.46%
CSELT	1,710.23	41.12%	8,336	71,598.92	1.53%	98.47%
COST266	11,866,362.78	96.42%	6,708	109,859.13	0.35%	99.65%

Now, let us take a closer look at the time required for solving CPP and CCP instances. Table 7.IV shows the average time consumed by the instances of CPP and CCP for each tested network. As for CPP, a very large number of problem instances is solved during the whole process and the average running times are very short, usually on the order of milliseconds. Let us recall that the size of a CPP instance relates to the number of connection requests in the original FIPP problem, which leads us to recognize the efficiency of the formulation proposed for CPP. In contrast, a small number of CCP instances is solved (one for each generated cycle plus one for proving optimality) and the average running times are a bit longer, but never above 26 seconds (in average). The short times required to solve the pricing problem at each iteration of the column generation algorithm explain why heuristic algorithms were ineffective in reducing the

Table 7.III: Performance of the column generation algorithm with pricing decomposition

Instances	# columns	# cycles	time (s)	master time	CPP time	CCP time
9N17S	206	26	2.56	2.34%	29.30%	68.36%
DFN-BWIN	19	19	0.70	2.86%	18.57%	78.57%
COST239	452	76	20.92	1.58%	23.18%	75.24%
POLSKA	321	24	4.21	3.09%	28.98%	67.93%
USA	561	45	14.17	2.82%	22.23%	74.95%
ATLANTA	470	20	6.00	8.67%	32.50%	58.83%
GERMANY	535	21	7.86	8.14%	25.06%	66.79%
NEWYORK	1,700	149	146.80	3.03%	17.72%	79.25%
EON-19	2,162	75	75.30	12.27%	32.58%	55.15%
TA1	1,668	74	68.12	6.93%	22.50%	70.57%
NORWAY	8,230	228	2,478.35	20.82%	12.96%	66.22%
BRAZIL	10,603	280	2,909.12	16.05%	8.58%	75.37%
EON-28	4,783	112	622.85	14.72%	16.08%	69.20%
BT	25,140	589	22,342.81	22.19%	9.88%	67.93%
CSELT	28,639	501	19,798.92	46.52%	10.07%	43.41%
COST266	31,019	441	21,056.27	29.82%	16.45%	53.73%

solution times.

7.7.2 Lower bound quality

The lower bounds obtained by the column generation algorithm for the tested networks are summarized in Table 7.V. For each network, we provide the working and protection costs as well as the redundancy ratio, which is a measure of architectural efficiency for survivable networks and is measured by the ratio of protection to working cost. As mentioned when presenting the formulations, we made the assumption that only mutually disjoint connections can be protected by the same cycle. The column generation generation proposed in [98], however, does not impose connection disjointness but it does not allow connections in a z-relationship with the cycle. In order to assess the impact of these assumptions on the obtained lower bounds, we tested both formu-

Table 7.IV: Average time required for solving an instance of CPP and CCP

Instances	CPP time (s)	CCP time (s)
9N17S	0.0010	0.06
DFN-BWIN	0.0007	0.03
COST239	0.0010	0.20
POLSKA	0.0013	0.11
USA	0.0011	0.23
ATLANTA	0.0012	0.17
GERMANY	0.0014	0.24
NEWYORK	0.0012	0.78
EON-19	0.0017	0.55
TA1	0.0017	0.64
NORWAY	0.0032	7.20
BRAZIL	0.0023	7.80
EON-28	0.0033	3.81
BT	0.0038	25.72
CSELT	0.0040	17.12
COST266	0.1210	22.58

lations on a 15-node family of related networks with number of links ranging from 16 to 30 [25]. The time limit of 10 hours was used for both formulations. Figure 7.5 illustrates the lower bounds found with the previous and the new formulations for each network. From the results, there is no clear advantage of any formulation, except for the largest networks, for which slightly better bounds were found with the new formulation. However, regarding the running times, the column generation proposed in this paper is remarkably superior to the previous one. Note that, for a fair comparison, the column generation algorithm without pricing decomposition was used in these experiments. Figure 7.6 shows how much time is consumed for running the column generation algorithm using both formulations. Now, it is clear that the new formulation is much more effective.

Table 7.V: Optimal solutions obtained by the column generation algorithm

Instances	working cost	protection cost	RR
9N17S	5,900.00	2,900.00	49.15%
DFN-BWIN	340,747.00	178,550.00	52.40%
COST239	137,170.00	60,263.91	43.93%
POLSKA	4,692,731.00	3,398,548.00	72.42%
USA	5,926,306.00	5,874,273.44	99.12%
ATLANTA	151,019.00	135,951.00	90.02%
GERMANY	407,130.00	446,372.50	109.64%
NEWYORK	1,437.00	485.93	33.82%
EON-19	91,273.12	89,489.39	98.05%
TA1	6,578,006.00	5,576,871.50	84.78%
NORWAY	10,792.00	5,504.75	51.01%
BRAZIL	2,689,967.00	1,905,574.40	70.84%
EON-28	1,796,669.00	1,908,707.50	106.24%
BT	6,444.00	2,752.69	42.72%
CSELT	4,159.00	1,710.23	41.12%
COST266	12,306,314.90	11,866,362.78	96.42%

7.7.3 Integer solution

As for the rounding algorithm, some parameters were defined according to an empirical analysis. Variable coefficients β_C represent the number of copies of cycle configuration C in the LP solution. More sophisticated criteria using information such as protection cost and the amount of protection provided by the cycles were also tested but less successful results were obtained with them. In addition, the maximum number of 100 iterations of the column generation algorithm is executed at each iteration of the rounding-based algorithm in order to keep the total computation time reasonable. In the round-robin algorithm, the list of cycle configuration is composed of all those obtained during the column generation algorithm and it is sorted according to the reduced costs in the optimal LP solution. Furthermore, the number of columns added at each iteration

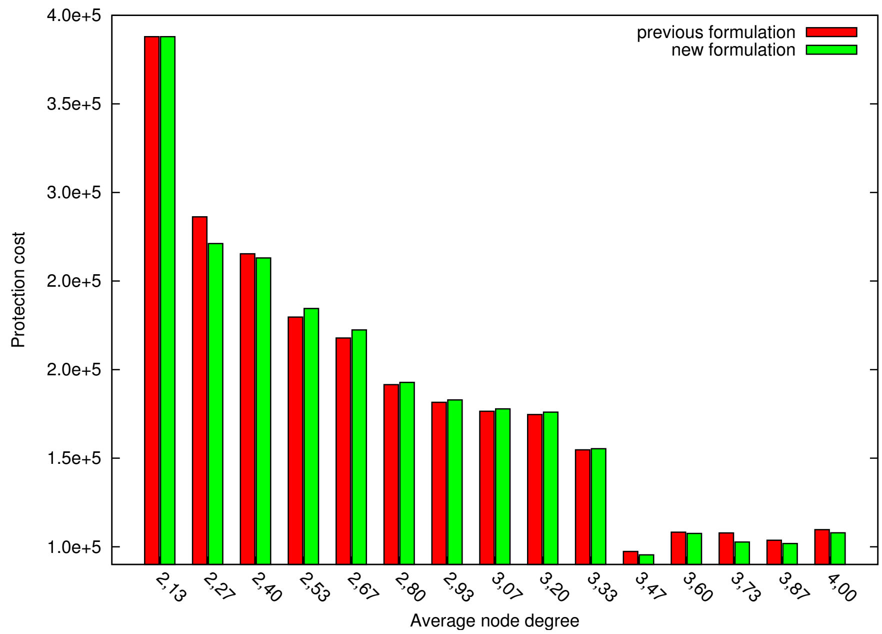


Figure 7.5: Different lower bounds obtained for a 15-node network family.

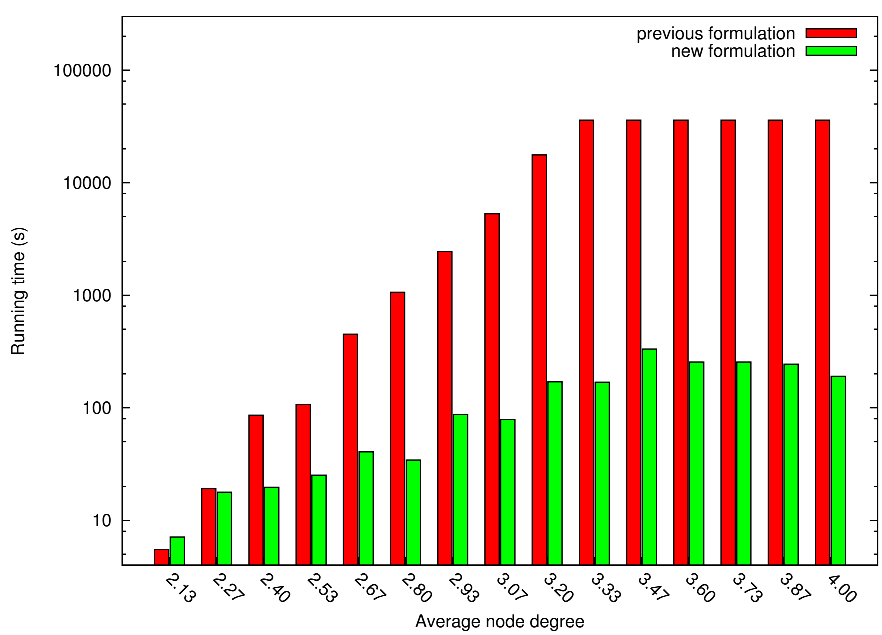


Figure 7.6: Running times for a 15-node network family.

was empirically set to 500.

Table 7.VI presents the results obtained by the two heuristics methods. Besides the protection cost, the table also reports the redundancy ratio, the gap between the optimal linear solution and the obtained integer solutions, and the running times for each tested network. Values shown in boldface indicate the winner algorithm with respect to the quality of solutions. Results show that the round robin algorithm is slightly more successful in obtaining good solution than the rounding-based algorithm, although optimality gap goes above 10% for one large instance. Both algorithms find near optimal solutions for several instances (optimality gaps within 1%). As for the running times, the round robin algorithm with the chosen parameter is significantly less time consuming.

7.8 Conclusion

In this paper, we proposed a new column generation approach for the efficient computation of FIPP p-cycles in survivable networks. The method relies on a new decomposition strategy in which two pricing problems are used, one for generating new cycles and another one for improving the use of existing cycles. By imposing mutual disjointness among working paths protected by the same cycle, we came up with a much more compact formulation for the pricing problem. This, together with the embedded decomposition, greatly reduced the running times and allowed us to approach problem instances of size never yet approached. Results have shown that this assumption did not affect the quality of the lower bounds obtained in comparison with an existing method, which in turn did not allow a cycle to protect requests in a z-relationship with the cycle.

Moreover, we proposed two heuristic methods to obtain integer solutions. The first one is a rounding-based algorithm and the second one consists in the iterative solution of small integer problems. Both algorithms were able to find near optimal solutions for several problem instances, with a slight advantage for the round robin algorithm. For few instances, less successful results were found by the heuristic methods.

As future research, it might be worthwhile to investigate valid inequalities to strengthen the formulation of the pricing problem thereby possibly accelerating the solution of the linear relaxation. Other research direction is the development of efficient heuristics in order to obtain good initial solutions for the column generation algorithm.

Table 7.VI: Results obtained by the proposed heuristic methods

Instances	Rounding				Round Robin			
	cost	RR	gap	time (s)	cost	RR	gap	time (s)
9N17S	**2,900**	49.15%	0.000%	2.96	**2,900**	49.15%	0.000%	0.01
DFN-BWIN	178,554	52.40%	0.002%	5.68	**178,553**	52.40%	0.002%	0.01
COST239	70,065	51.08%	16.264%	68.21	**64,500**	47.02%	7.029%	0.56
POLSKA	3,401,187	72.48%	0.078%	36.59	**3,399,550**	72,44%	0,029%	0.05
USA	5,939,776	100.23%	1.115%	88.44	**5,896,122**	99.49%	0.372%	0.09
ATLANTA	**135,951**	90.02%	0.000%	23.72	135,980	90.04%	0.021%	0.01
GERMANY	448,131	110.07%	0.394%	75.74	**446,595**	109.69%	0.050%	0.07
NEWYORK	514	35.77%	5.777%	759.4	**505**	35.14%	3.925%	2.37
EON-19	90,504.83	99.16%	1.135%	204.79	**89,585.942**	98.15%	0.108%	10.29
TA1	**5,576,893**	84.78%	0.000%	751.48	5,576,928	84.78%	0.001%	0.20
NORWAY	5,716	52.97%	3.838%	15,406.41	5,686	52.69%	3.293%	1,020.64
BRAZIL	**1,983,248**	73.73%	4.076%	10,333.65	1,992,300	74.06%	4.551%	101.46
EON-28	1,995,875	111.09%	4.567%	3,309.48	**1,972,044**	109.76%	3.318%	344.1
BT	**2,972**	46.12%	7.967%	119,404.94	3,025	46.94%	9.893%	55,822.07
CSELT	**1,834**	44.10%	7.237%	69,910.77	1,935	46.53%	13.143%	43,896.45
COST266	**11,866,638.15**	96.43%	0.002%	153,654.73	11,868,033.82	96.44%	0.014%	61.74
Average:	1,981,297.44	73.10%	3.28%	23,377.31	1,977,140.17	72.80%	2.86%	6,328.76

CHAPTER 8

LOWER BOUNDS FOR THE DESIGN OF P-CYCLE NETWORKS WITH AVAILABILITY REQUIREMENTS

8.1 Chapter presentation

This chapter concerns a study on a first column generation formulation and solution scheme for the availability-aware design of survivable networks based on link-protecting p-cycles. Herein, we present some preliminary results obtained for small data instances and compare them with existing results.

8.2 Introduction

The method of pre-configured protection cycles (p-cycles) [39, 46] has emerged as a topic of great importance over the past few years due to its capabilities of achieving ring-like high speed protection with mesh-like high efficiency in the use of spare capacity. The objective of this and other protection schemes has been to guarantee the restoration of affected services in the event of any single element failures. The spare capacity placement method is usually referred to as the method for determining the amount of spare capacity that must be provisioned in the network to meet the requirement of full survivability of any single failure.

However, using these protection methods to make a network fully restorable for single failures is not a guarantee that the availability of the service in the occurrence of higher order failures will be 100%. Several approaches have therefore been designed to improve the robustness of high capacity mesh transport networks against dual failures. These approaches have either considered (pre-failure) strategies for addition of further protection capacity to achieve full [114] or partial dual-failure survivability [15, 18] or have assumed reconfiguration of protection resources after the occurrence of the first failure to better withstand future failures [70, 111]. More recently, the authors of [44] have argued that, in addition to the above mentioned approaches, reductions in the physical repair time of failures (i.e., shorter outage periods) can also enhance service availability. They showed that an economic strategy exists for balancing the tradeoffs between capacity investment and Mean Time To Repair (MTTR) reduction efforts to achieving high service availability in networks designed to be 100% restorable against single failures.

The authors of [16] have studied the availability in link-restorable mesh networks. The availability analysis is based on the computational analysis of the restorability of a network to all possible dual-failure scenarios and the authors explained the relation between the path availability and the service restorability. In [17], the authors developed an analytical expression for the availability of paths in networks using p-cycles as the protection mechanism. The model presented is based on the calculation of the unavailability caused by the effects of dual failures and the authors have used the concept of *cutset method* or *protection domain* to determine the service availability. Two new models were introduced for simultaneous path routing and capacity design for p-cycle networks that serve a class with higher availability requirements in addition to the traditional single-failure protected class. An availability-aware service provisioning method in p-cycle based mesh networks is presented in [85]; therein, the end-to-end service availability is analytically derived as a function of the link unavailability, using the concept of protection domain. The spare capacity is allocated, through a non-joint optimization model, to meet the availability requirements of the end-to-end traffic.

The work in [61] aimed to address some shortcomings which make the model and analysis reported in [85] inaccurate, and hence, to propose a more elaborate model, termed as ApC model. Concerning the availability analysis, the authors thoroughly enumerate all dual-failure scenarios which may lead to an outage on the path through which the service is routed. Then, they show that a very careful analysis must be done on each protection domain traversed by the service paths so as to avoid an overestimation of the unavailability. As a result, the proposed ApC model is more accurate but less scalable than that proposed in [85]. Therefore, the authors also propose some techniques to address the scalability issues of the ApC model, which also results in a smaller overestimation than in [85]. For instance, they do not consider all cycles in the network but only a set of candidate cycles when solving the proposed ILP model.

Although more accurate solutions are found in [61], the solution approaches proposed therein remain undoubtedly heuristic and an overestimation of unavailability is still obtained. Consequently, better solutions are likely to exist. Thus, the objective of the study carried out in this chapter is to find lower bounds for optimal solutions and hopefully obtain better solutions than those obtained in [61]. For that, we propose a new ILP formulation for the problem and solve its linear relaxation using a column generation algorithm which employs the same decomposition principle proposed in Chapter 7.

The rest of this chapter is structured as follows. In Section 8.3, we present a thorough analysis of the unavailability for a working path in a p-cycle-based network, which was introduced in [61]. In Section 8.4, we present a new ILP formulation and suggest its solution with a column generation algorithm. In Section 8.5, we describe the pricing problem and present an formulation for it as well as an embedded decomposition method. In Section 8.6, we describe our computational experiments and discuss the obtained results.

8.3 Availability analysis of p-cycle based networks

The service availability is defined as *the probability of the system being found in the operating state at some time* t *in the future given that the system started in the operating state at time* $t = 0$ [38]. The availability of a service path is influenced by many factors such as the statistics of network element failures, repair time, mean restoration time, etc.

One of the most common and practical approaches for finding service availability in a network is the "cutset method" where the failures that cause service outage are divided into non-overlapping categories and a dual-failure can only belong to one of these categories [85]. Such an analysis assumes that each link has the same physical unavailability (U).

The "protection domain" concept associates a p-cycle p and a particular service path k and it is defined as the set of links (on-cycle or straddling) of path k which are protected by p [85]. Accordingly, a p-cycle p providing a non-empty protection domain for a request routed along working path k is partitioned into four mutually exclusive subsets, as enumerated below:

O_k^p Set of links on cycle p which are also on working path k and protected by p.

$O_{\overline{k}}^p$ Set of links on cycle p which are not on working path k and also those on-cycle links traversed by path k but not protected by p.

S_k^p Set of straddling links which are on working path k and protected by p.

$S_{\overline{k}}^p$ Set of all straddling links protected by p (with respect to other working paths), except those in S_k^p.

Note that the straddling links which are not protected by p are not of interest for the unavailability analysis of any service path. Indeed, a failure on any of those links will not affect the restorability of path k.

At this point, we are able to identify the categories of dual failure scenarios that may lead to outage on working path k in a protection domain and to define the expression of the unavailability contribution due to each category. To do so, let us denote links ℓ_1 and ℓ_2 as the failed links. Also, let us assume that the physical unavailability of each link in the network is equal to U. Thus, the failure categories are enumerated as follows:

(C_1) Dual failure scenario where $\ell_1 \in O_k^p$ and $\ell_2 \in O_{\overline{k}}^p$. The order in which the failures occur is not important because in both cases there may be a service outage on path k. Therefore, the unavailability due to the dual failures in this category is given by expression (8.1):

$$U_{C_1} = |O_k^p| \cdot |O_{\overline{k}}^p| \cdot U^2. \tag{8.1}$$

(C_2) Dual failure scenario where $\ell_1 \in O_k^p$ and $\ell_2 \in S_{\overline{k}}^p$. In this case, the order of the failures is important: There may be a service outage only if the first failure occurs on link ℓ_2, assuming that this link is *fully loaded*[1]. Since link ℓ_2 fails before ℓ_1 with a 0.5 probability, the unavailability contribution of this type of dual failures is expressed by (8.2):

$$U_{C_2} = \frac{1}{2}|O_k^p| \cdot |S_{\overline{k}}^p| \cdot U^2. \tag{8.2}$$

1. A straddling link is called *fully loaded* when two units of working capacity over it are protected by a unit capacity cycle.

(C_3) Dual failure scenario where $\ell_1 \in O_k^p$ and $\ell_2 \in S_k^p$. Upon such dual failures, there may be a service outage regardless of the order in which the failures occur and the resulting unavailability is expressed as:

$$U_{C_3} = |O_k^p| \cdot |S_k^p| \cdot U^2. \tag{8.3}$$

(C_4) Dual failure scenario where $\ell_1 \in S_k^p$ and $\ell_2 \in O_{\bar{k}}^p$. If link $\ell_2 \in O_{\bar{k}}^p$ fails first, this failure sequence will definitely lead to service outage on path k. However, when the first failure occurs on ℓ_1, there may be an outage only if ℓ_2 is traversed by the cycle half assigned to protect ℓ_1. Assuming that there is a 0.5 probability of the assigned half to be affected and since each failure permutation has an equal probability of 0.5, this type of failures contributes to service unavailability according to expression (8.4):

$$U_{C_4} = \frac{3}{4}|S_k^p| \cdot |O_{\bar{k}}^p| \cdot U^2. \tag{8.4}$$

(C_5) Dual failure scenario where both ℓ_1 and $\ell_2 \in S_k^p$. Assuming again that these straddling links are fully loaded, these dual failures may cause an outage on path k and the resulting unavailability is given by expression (8.5):

$$U_{C_5} = \frac{1}{2}|S_k^p| \cdot (|S_k^p| - 1) \cdot U^2. \tag{8.5}$$

(C_6) Dual failure scenario where $\ell_1 \in S_k^p$ and $\ell_2 \in S_{\bar{k}}^p$. In this category, there may be service outage only if links ℓ_2 fails first. Therefore, the contribution of this failure scenario for unavailability of path k is given by:

$$U_{C_6} = \frac{1}{2}|S_k^p| \cdot |S_{\bar{k}}^p| \cdot U^2. \tag{8.6}$$

(C_7) Dual failure scenario where both ℓ_1 and $\ell_2 \in O_k^p$. In this case,

$$U_{C_7} = \frac{1}{2}|O_k^p| \cdot (|O_k^p| - 1) \cdot U^2. \tag{8.7}$$

Since the failure categories are mutually exclusive, the probability of service outage working path k in protection domain d can be obtained by summing the unavailability due to each category:

$$U_d = \sum_{i=1}^{7} U_{C_i}. \tag{8.8}$$

Moreover, since the protection domains with respect to a given route are in series [38], the unavailability of a path k in a p-cycle based network can be approximated as the sum of the unavailability in the different crossed protection domains:

$$U_k \approx \sum_{d \in \mathcal{D}(k)} U_d \tag{8.9}$$

where $\mathcal{D}(k)$ is the set of protection domains traversed by path k. The inaccuracy in this expression arises from the fact that higher degree polynomials of U_d have been neglected. Such an approximation is reasonable, as the service unavailability is usually very small and higher degree terms are consequently negligible.

8.3.1 An illustrative example

For ease of understanding, let us present the following example, which is illustrated in Figure 8.1. In this example, there are two working paths and two different p-cycles. For working path k_1 (A-C-G-F), we assume that links C-G and G-F are protected by p-cycle p_1 (on the left) and link A-C by p-cycle p_2 (on the right). Therefore, path k_1 traverses two protection domains. Note that, although link A-C straddles p_1, it should not be considered in this protection domain since it is being protected by p_2. In contrast, we assume that working path k_2 (D-C-G) is completely protected by p-cycle p_1.

Following with this example, Table 8.I shows the sets of links used to compute unavailability for each path. Then, according to expressions (8.1)-(8.7), we have that the unavailability of both paths k_1 and k_2 in the protection domains associated with cycle p_1 is equal to $13.25U^2$, while protection domain associated with cycle p_2 contributes to an additional unavailability of $3.75U^2$ for path k_1. Thus, the total probability of a service outage on paths k_1 and k_2 is $17U^2$ and $13.25U^2$, respectively.

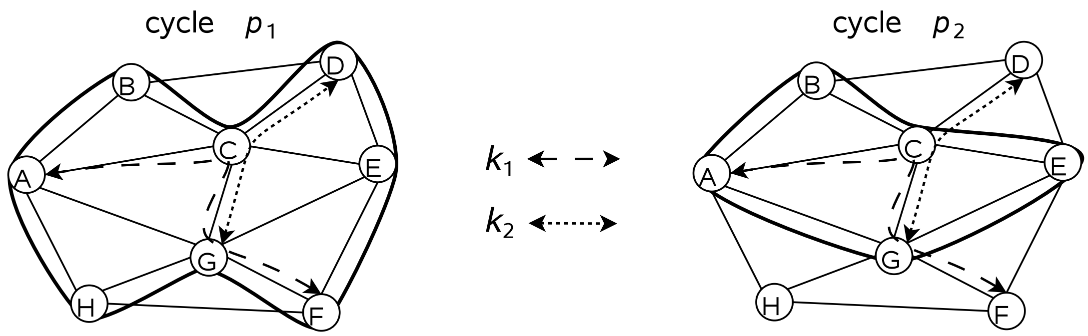


Figure 8.1: An example of protection domains.

Table 8.I: Partition of links into sets for the example in Figure 8.1.

		path k_1	path k_2
cycle p_1	O_k^p	{G-F}	{C-D}
	$O_{\overline{k}}^p$	{A-B, B-C, C-D, D-E, E-F, G-H, H-A}	{A-B, B-C, D-E, E-F, F-G, G-H, H-A}
	S_k^p	{C-G}	{C-G}
	$S_{\overline{k}}^p$	$\emptyset$	$\emptyset$
cycle p_2	O_k^p	$\emptyset$	–
	$O_{\overline{k}}^p$	{A-B, B-C, C-E, E-G, G-A}	–
	S_k^p	{A-C}	–
	$S_{\overline{k}}^p$	$\emptyset$	–

8.3.2 Problem definition

In this section, we formally define the availability-aware p-cycle-based network design problem (APNDP). We consider an optical network represented by an undirected graph $G = (V, L)$ where V is the set of nodes and L is the set of links, which represent physical entities that collect all channels between neighbor nodes. For each link $\ell \in L$, there is a linear cost c_ℓ for using one unit of spare capacity on that link. Span costs can represent information such that cost of interconnection equipment at endpoints, link length, etc. Furthermore, we are given a set of unit working paths for connection requests between pairs of nodes, denoted by K. Note that a multi-unit request can be easily transformed into individual unit requests between the same pair of nodes. Let $\mathcal{P}$ denote the set of all candidate cycles in the network. The cost of cycle $p \in \mathcal{P}$ is given by $\text{COST}_p = \sum_{\ell \in p} c_\ell$.

Given maximum unavailability values MU_k, $\forall k \in K$, the APNDP consists in determining a subset of link protection p-cycles to fully protect all working paths in K while minimizing the overall protection cost and respecting the maximum unavailability of each working path. The overall protection cost is given by $\sum_{p \in \mathcal{P}} \text{COST}_p v_p$, where v_p is the number of unit copies of cycle p. In this study, we assume that all working paths require the same maximum unavailability MU.

8.4 A mathematical formulation for APNDP

In the following, we present an ILP formulation for APNDP. The decision variables are defined with respect to protection domains offered by the cycles. Indeed, each variable corresponds to a set of protection domains, each one traversed a different working path, in a particular cycle. For each cycle $p \in \mathcal{P}$, let us define the set $\mathcal{D}_p$ where each element $D \in \mathcal{D}_{\mathcal{P}}$ is a different set of protection domains and let us also define D_k as the protection domain in D traversed by working path k. Thus, let us define binary variable $n_D, D \in \mathcal{D}_{\mathcal{P}}, p \in \mathcal{P}$, such that $n_D = 1$ if and only if particular set D of protection domains in cycle p is chosen to contribute to protection of the working paths. For a cycle $p \in \mathcal{P}$ and a set of protection domains $D \in \mathcal{D}_{\mathcal{P}}$, let us further define the following parameters:

$\text{COST}_D =$ $\text{COST}_p v_D$, where v_D is the number of copies of cycle p used in D.

$\alpha_{k\ell}^D =$ 1 if and only if link ℓ of path k is protected in D_k.

$u_k^D =$ probability of service outage on path k due to protection domain D_k.

The APNDP can be then formulated as follows:

$$\text{Minimize} \sum_{p \in \mathcal{P}} \sum_{D \in \mathcal{D}_p} \text{COST}_D \, n_D \tag{8.10}$$

subject to:

$$(\pi_{k\ell}) \qquad \sum_{p\in\mathcal{P}} \sum_{D\in\mathcal{D}_p} \alpha_{k\ell}^{D}\, n_D \geq 1 \qquad \forall k \in K, \ell \in k \tag{8.11}$$

$$(\mu_k) \qquad \sum_{p\in\mathcal{P}} \sum_{D\in\mathcal{D}_p} (u_k^D\, U^2)\, n_D \leq MU \qquad \forall k \in K \tag{8.12}$$

$$(\lambda_p) \qquad \sum_{D\in\mathcal{D}_p} n_D \leq 1 \qquad \forall p \in \mathcal{P} \tag{8.13}$$

$$n_D \in \{0,1\} \qquad \forall D \in \mathcal{D}_p, p \in \mathcal{P}. \tag{8.14}$$

The objective function (8.10) calculates the total protection cost. Constraints (8.11) ensure that every end-to-end working path is fully protected over all protection domains. Constraints (8.12) ensure that the maximum probability of service outage on every path is not exceeded. Since value U is constant, these constraints can be rewritten as:

$$\sum_{p\in\mathcal{P}} \sum_{D\in\mathcal{D}_p} u_k^D\, n_D \leq \frac{MU}{U^2}. \tag{8.15}$$

According to the definition of protection domain, we have inequalities (8.13) stating that at most one protection domain in each cycle is selected for each working path. Otherwise, the unavailability values would not be properly calculated.

The major difficulty with the APNDP model is clearly the huge number of possible sets of protection domains. Hence, we use a column generation algorithm for solving the APNDP model, thereby taking all columns into account implicitly. In the next section, we describe and formulate the resulting pricing problem.

8.5 The pricing problem

The pricing problem for the APNDP model consist in identifying the cycle and the set of protection domains which corresponds to the variable with minimum reduced cost.

The reduced cost of variable $n_D, D \in \mathcal{D}_p$, is expressed as follows:

$$\overline{\text{COST}}_D = \text{COST}_D - \sum_{k \in K} \sum_{\ell \in k} \pi_\ell^k \, \alpha_{k\ell}^D - \sum_{k \in K} \mu_k \, u_k^D - \lambda_p \tag{8.16}$$

where $\pi_\ell^k \geq 0$, $\mu_k \leq 0$, and $\lambda_p \leq 0$ are the dual variables obtained respectively from constraints (8.11), (8.12), and (8.13) in the APNDP model.

In Chapter 7, we proposed a hierarchical decomposition of the pricing problem. This decomposition produces two different pricing problems that can be solved for finding negative reduced columns. Here, we apply the same decomposition strategy which gives us the following problems:

Pricing problem 1. Determine the cycle as well as the set of protection domains which leads to the column with minimum reduced cost.

Pricing problem 2. For a given cycle, determine the set of protection domains yielding a minimum reduced cost.

In this study, the column generation algorithm proceeds in the same way as described in Section 7.5. More particularly, at each iteration, we first try to find an improving column by solving Pricing problem 2 for each known cycle, if there is any. If no cycle is known or no improving column can be found, we then solve Pricing problem 1 which will either provide an improving column or prove that optimality has been reached.

In the following, we present the mathematical formulations for both pricing problems.

8.5.1 Pricing problem 1

Firstly, let us address the pricing problem for finding a column composed of a new cycle and a set of protection domains with minimum reduced cost. The proposed ILP formulation for this problem consists of the following variables:

$x_\ell \in \{0, 1\}$, with $x_\ell = 1$ if and only if link ℓ belongs to the cycle.

$z_\ell \in \{0, 1\}$, with $z_\ell = 1$ if link ℓ has both end nodes on the cycle.

$y_i \in \{0, 1\}$, with $y_i = 1$ if and only if node i is traversed by the cycle.

$w_\ell \in \{0,1\}$, with $w_\ell = 1$ if link ℓ straddles the cycle and there is at least one channel protected in this link.

$v_\ell \in \mathbb{Z}_+$, represents the capacity to be installed on link ℓ.

$v \in \mathbb{Z}_+$, represents the number of used copies of the cycle.

$\alpha^{\text{ON}}_{k\ell} \in \{0,1\}$, with $\alpha^{\text{ON}}_{k\ell} = 1$ if and only if link ℓ of path k is protected and *on* the cycle.

$\alpha^{\text{ST}}_{k\ell} \in \{0,1\}$, with $\alpha^{\text{ST}}_{k\ell} = 1$ if and only if link ℓ of path k is protected and *straddling* the cycle.

$u^{\text{ON}}_{k\ell} \in \mathbb{R}_+$, represents the number of all combinations of dual failures involving link ℓ of path k if link ℓ is protected and on cycle; 0, otherwise.

$u^{\text{ST}}_{k\ell} \in \mathbb{R}_+$, represents the number of all combinations of dual failures involving link ℓ of path k, if link ℓ is protected and straddles the cycle; 0, otherwise.

Variables $u^{\text{ON}}_{k\ell}$ and $u^{\text{ST}}_{k\ell}$ represent the individual contribution of link ℓ to the unavailability of k in the protection domain. In particular, if $\ell \in O^p_k$ then $u^{\text{ON}}_{k\ell} > 0$ and $u^{\text{ST}}_{k\ell} = 0$; if $\ell \in S^p_k$ then $u^{\text{ON}}_{k\ell} = 0$ and $u^{\text{ST}}_{k\ell} > 0$; otherwise, $u^{\text{ON}}_{k\ell} = u^{\text{ST}}_{k\ell} = 0$. Using the example shown in Figure 8.1, link G-F of path k_1 in the protection domain given by cycle p_1 belongs $O^{p_1}_{k_1}$ and variable $u^{\text{ON}}_{k\ell}$ would assume value 8. In contrast, link C-G of path k_1 in the same protection domain belongs to $S^{p_1}_{k_1}$ and the value of variable $u^{\text{ST}}_{k\ell}$ would be 5.25. The formula to obtain these values are given later in this section. Note that the sum of these values corresponds exactly to number 13.25 in the unavailability of path k_1 in this protection domain, i.e., $13.25U^2$.

Let us define $\delta(S), S \subset V$, as the cut induced by S, i.e, the set of links incident to a node in S and another node in $V \backslash S$. For a single node $i \in V$, we denote $\delta(i) = \delta(\{i\})$. Additionally, let $L(S), S \subset V$, be the set of links induced by S, i.e., the set of links whose both adjacent nodes belong to S. Pricing problem 1 is formulated as follows:
Minimize

$$\sum_{\ell \in L} c_\ell \, v_\ell - \sum_{k \in K} \sum_{\ell \in k} \pi_{k\ell} \, (\alpha^{\text{ON}}_{k\ell} + \alpha^{\text{ST}}_{k\ell}) + \sum_{k \in K} \left[\mu_k \sum_{\ell \in k} (u^{\text{ON}}_{k\ell} + u^{\text{ST}}_{k\ell}) \right] \tag{8.17}$$

subject to:

$$\sum_{\ell \in \omega(i)} x_\ell = 2y_i \quad \forall i \in V \tag{8.18}$$

$$\sum_{\ell \in \delta(V')} x_\ell \geq 2(x_{\ell'} + x_{\ell''} - 1) \quad \forall S \subset V, 2 < |S| < |V| - 2,$$

$$\ell' \in L(S), \ell'' \in L(V \setminus S) \tag{8.19}$$

$$z_\ell \leq y_i \quad \forall i \in V, \ell \in \delta(i) \tag{8.20}$$

$$z_\ell \geq y_i + y_j - 1 \quad \forall \ell = \{i, j\} \in L \tag{8.21}$$

$$\alpha_{k\ell}^{\text{ON}} \leq x_\ell \quad \forall k \in K, \ell \in k \tag{8.22}$$

$$\alpha_{k\ell}^{\text{ST}} \leq z_\ell - x_\ell \quad \forall k \in K, \ell \in k \tag{8.23}$$

$$\sum_{k \in K: \ell \in k} \alpha_{k\ell}^{\text{ON}} \leq v \quad \forall \ell \in L \tag{8.24}$$

$$\sum_{k \in K: \ell \in k} \alpha_{k\ell}^{\text{ST}} \leq 2v \quad \forall \ell \in L \tag{8.25}$$

$$v + (x_\ell - 1)M \leq v_\ell \quad \forall \ell \in L \tag{8.26}$$

$$\alpha_{k\ell}^{\text{ST}} \leq w_\ell \quad \forall k \in K, \ell \in k \tag{8.27}$$

$$\sum_{k \in K: \ell \in k} \alpha_{k\ell}^{\text{ST}} \geq w_\ell \quad \forall \ell \in L \tag{8.28}$$

$$(1 - \alpha_{k\ell}^{\text{ON}})M + u_{k\ell}^{\text{ON}} \geq f^{\text{ON}}(k, \ell) \quad \forall k \in K, \ell \in k \tag{8.29}$$

$$(1 - \alpha_{k\ell}^{\text{ST}})M + u_{k\ell}^{\text{ST}} \geq f^{\text{ST}}(k, \ell) \quad \forall k \in K, \ell \in k \tag{8.30}$$

$$x_\ell, z_\ell, w_\ell \in \{0, 1\} \quad \forall \ell \in L \tag{8.31}$$

$$y_i \in \{0, 1\} \quad \forall i \in V \tag{8.32}$$

$$\alpha_{k\ell}^{\text{ON}}, \alpha_{k\ell}^{\text{ST}} \in \{0, 1\} \quad \forall k \in K, \ell \in k \tag{8.33}$$

$$u_{k\ell}^{\text{ON}}, u_{k\ell}^{\text{ST}} \in \mathbb{R}_+ \quad \forall k \in K, \ell \in k \tag{8.34}$$

$$v_\ell \in \mathbb{Z}_+ \quad \forall \ell \in L \tag{8.35}$$

$$v \in \mathbb{Z}_+ \tag{8.36}$$

The objective function (8.17) calculates the reduced cost as previously described by expression (8.16). Since $\pi_{k\ell} \geq 0$ and $\mu_k \leq 0$, a good column corresponds to a low cost p-cycle and a set of "profitable" protection domains leading to low unavailability. Coefficients $\alpha_{k\ell}^{D}$ in the master problem are obtained from the sum of the corresponding variables $\alpha_{k\ell}^{\text{ON}}$ and $\alpha_{k\ell}^{\text{ST}}$ in the pricing problem. Moreover, coefficients u_k^D obtained by calculating the following sum: $\sum_{\ell \in k}(u_{k\ell}^{\text{ON}} + u_{k\ell}^{\text{ST}})$. Remark that, instead of calculating the total unavailability of a service path in a protection domain, this formulation calculates the individual contribution of each link of the path to its unavailability.

We recall that Pricing Problem 1 is solved in order to find an improving column associated with an unknown cycle, say p. Since, the dual variable λ_p present in the reduced cost of such column is equal to zero, this term does not appear in objective function (8.17) .

The degree constraints (8.18) require the degree of each node to be either 0 or 2. Inequalities (8.19) are connectivity constraints stating that each cut separating two selected nodes must be crossed twice. Constraints (8.20) and (8.21) identify the links whose both end nodes are crossed the cycle.

Variables $\alpha_{k\ell}^{\text{ON}}$ ($\alpha_{k\ell}^{\text{ST}}$) can assume value 1 only if link ℓ is on (straddling) the cycle, which is ensured by constraints (8.22) (constraints (8.23), in the straddling case). Constraints (8.24) and (8.25) determine the number of cycle copies needed to provide the required protection. Constraints (8.26) determine the number of channels to be installed on each link. Constraints (8.27) and (8.27) identify whether a link straddles the cycle and there is at least one protected channel on it.

Constraints (8.29) and (8.30) calculate the number of dual failure sequences involving link ℓ of path k whether ℓ is an on-cycle or a straddling link, respectively. Since this is a minimization problem, variables $u_{k\ell}^{\text{ON}}$ and $u_{k\ell}^{\text{ST}}$ are forced to be as low as possible. Therefore, if link ℓ of a working path is protected, the number of dual failure sequences involving ℓ is correctly calculated, assuming that the value of constant M is properly set. The right end sides of these constraints are derived from the study on availability carried out in Section 8.3. Accordingly, if link $\ell \in k$ is protected and on the cycle, the dual failure sequences involving ℓ belong to either category C_1, C_2, C_3 or C_7. The number of the dual failure sequences involving on-cycle link ℓ ($u^{\text{ON}}_{k\ell}$) is then given by:

$$f^{\text{ON}}(k,\ell) = \overbrace{\sum_{\ell' \in L} x_{\ell'} - \sum_{\ell' \in k} \alpha^{\text{ON}}_{k\ell'}}^{C_1} + \overbrace{\frac{1}{2}\left[\sum_{\ell' \in L} w_{\ell'} - \sum_{\ell' \in k} \alpha^{\text{ST}}_{k\ell'}\right]}^{C_2} + \overbrace{\sum_{\ell' \in k} \alpha^{\text{ST}}_{k\ell'}}^{C_3} + \overbrace{\frac{1}{2}\left[\sum_{\ell' \in k} \alpha^{\text{ON}}_{k\ell'} - 1\right]}^{C_7}.$$

On the other hand, if link $\ell \in k$ is protected and straddling the cycle, the dual failure sequences involving this link belong to either category C_3, C_4, C_5 or C_6. However, since the dual failures in category C_3 are already taken into account when computing $u^{\text{ON}}_{k\ell'}$ for another link $\ell' \in k$, these dual failures are not considered in $u^{\text{ST}}_{k\ell'}$. The number of dual failure sequences involving straddling link ℓ ($u^{\text{ST}}_{k\ell}$) is given by:

$$f^{\text{ST}}(k,\ell) = \overbrace{\frac{3}{4}\left[\sum_{\ell' \in L} x_{\ell'} - \sum_{\ell' \in k} \alpha^{\text{ON}}_{k\ell'}\right]}^{C_4} + \overbrace{\frac{1}{2}\left[\sum_{\ell' \in k} \alpha^{\text{ST}}_{k\ell'} - 1\right]}^{C_5} + \overbrace{\frac{1}{2}\left[\sum_{\ell' \in L} w_{\ell'} - \sum_{\ell' \in k} \alpha^{\text{ST}}_{k\ell'}\right]}^{C_6}.$$

8.5.2 Pricing problem 2

We now present an ILP formulation for the pricing problem of finding a set of protection domains for a known cycle. Let p denote the given cycle and $\hat{p}$ denote the set of links straddling p. Dual price λ_p is constant and therefore can be omitted in the objective function. Thus, Pricing problem 2 can be formulated as follows:
Minimize

$$\sum_{\ell} \text{COST}_p\, v - \sum_{k \in K}\left[\sum_{\ell \in k \cap p} \pi_{k\ell}\, \alpha^{\text{ON}}_{k\ell} + \sum_{\ell \in k \cap \hat{p}} \pi_{k\ell}\, \alpha^{\text{ST}}_{k\ell}\right] + \sum_{k \in K} \mu_k \left[\sum_{\ell \in k \cap p} u^{\text{ON}}_{k\ell} + \sum_{\ell \in k \cap \hat{p}} u^{\text{ST}}_{k\ell}\right] \tag{8.37}$$

subject to:

$$\sum_{k \in K : \ell \in k} \alpha^{\text{ON}}_{k\ell} \leq \; v \quad \forall \ell \in p \tag{8.38}$$

$$\sum_{k \in K : \ell \in k} \alpha^{\text{ST}}_{k\ell} \leq \; 2v \quad \forall \ell \in \hat{p} \tag{8.39}$$

$$\alpha^{\text{ST}}_{k\ell} \leq w_\ell \quad \forall k \in K, \ell \in k \cap \hat{p} \tag{8.40}$$

$$\sum_{k \in K : \ell \in k} \alpha^{\text{ST}}_{k\ell} \geq w_\ell \quad \forall \ell \in \hat{p} \tag{8.41}$$

$$(1 - \alpha^{\text{ON}}_{k\ell})M + u^{\text{ON}}_{k\ell} \geq \; f^{\text{ON}}(k, \ell) \quad \forall k \in K, \ell \in k \cap p \tag{8.42}$$

$$(1 - \alpha^{\text{ST}}_{k\ell})M + u^{\text{ST}}_{k\ell} \geq \; f^{\text{ST}}(k, \ell) \quad \forall k \in K, \ell \in k \cap \hat{p} \tag{8.43}$$

$$w_\ell \in \{0, 1\} \quad \forall \ell \in \hat{p} \tag{8.44}$$

$$\alpha^{\text{ON}}_{k\ell} \in \{0, 1\}, u^{\text{ON}}_{k\ell} \in \mathbb{R}_+ \quad \forall k \in K, \ell \in k \cap p \tag{8.45}$$

$$\alpha^{\text{ST}}_{k\ell} \in \{0, 1\}, u^{\text{ST}}_{k\ell} \in \mathbb{R}_+ \quad \forall k \in K, \ell \in k \cap \hat{p} \tag{8.46}$$

$$v \in \mathbb{Z}_+ \tag{8.47}$$

It should be noted that there is no constraint on the cycle structure in this model. Objective function (8.37) is equivalent to (8.17) as well as constraints (8.38)-(8.43) are equivalent to their counterparts (8.24)-(8.30), with the appropriate modifications to take into account that the cycle is known beforehand. This is responsible for a very significant reduction in terms of complexity in comparison with Pricing problem 1.

The number of dual failures involving link $\ell \in k$ whether ℓ is on ($u^{\text{ON}}_{k\ell}$) or straddling ($u^{\text{ST}}_{k\ell}$) the cycle is now given respectively by:

$$f^{\text{ON}}(k, \ell) = \overbrace{|p| - \sum_{\ell' \in k \cap p} \alpha^{\text{ON}}_{k\ell'}}^{\text{C1}} + \overbrace{\frac{1}{2}\left[\sum_{\ell' \in \hat{p}} w_{\ell'} - \sum_{\ell' \in k \cap \hat{p}} \alpha^{\text{ST}}_{k\ell'}\right]}^{\text{C2}} + \overbrace{\sum_{\ell' \in k \cap \hat{p}} \alpha^{\text{ST}}_{k\ell'}}^{\text{C3}} + \overbrace{\frac{1}{2}\left[\sum_{\ell' \in k \cap p} \alpha^{\text{ON}}_{k\ell'} - 1\right]}^{\text{C7}}.$$

and

$$f^{\text{ST}}(k,\ell) = \overbrace{\frac{3}{4}\left[|p| - \sum_{\ell' \in k \cap p} \alpha^{\text{ON}}_{k\ell'}\right]}^{\text{C4}} + \overbrace{\frac{1}{2}\left[\sum_{\ell' \in k \cap \hat{p}} \alpha^{\text{ST}}_{k\ell'} - 1\right]}^{\text{C5}} + \overbrace{\frac{1}{2}\left[\sum_{\ell' \in \hat{p}} w_{\ell'} - \sum_{\ell' \in k \cap \hat{p}} \alpha^{\text{ST}}_{k\ell'}\right]}^{\text{C6}}.$$

8.6 Results and discussion

We evaluate our column generation (CG) algorithm on different network scenarios and compare the obtained lower bounds with results obtained in [61]. For that, we use two network topologies: 9N17S, with 9 nodes and 17 links [38], and COST239, with 11 nodes and 26 links [6]. In all cases, we assume that all demands are routed *a priori* using Dijkstra's shortest path algorithm and that the traffic instances are composed of two unit requests between each pair of nodes. We also assume that each link has enough spare channels to support the protection capacity required by the optimal solution. The probability of physical unavailability for each link (U) is assumed to be equal to 10^{-3}. The CG algorithm was implemented in C++ programming language using CPLEX Concert Technology library and version 10.1 of CPLEX solver. The computational experiments were performed on an AMD 64-bit machine with 16GB of RAM.

In order to assess the quality of the lower bounds obtained by our CG algorithm, we compare them with the integer solutions provided in [61] by using the same instances and availability limits. Table 8.II reports the cost redundancy which is achieved in the results for two network instances. The first column shows the minimum availability required by every unit request in the network. The second and third columns present the results obtained in [61] and by the CG algorithm, respectively, followed by the gaps between these values. Our CG algorithm was not able to provide results for larger availability values than those presented in Table 8.II, especially for COST239 network. Consequently, we will not be able to compare all obtained values against those from [61].

We can see that there is not a very large gap between the integer solutions from [61] and our lower bounds for 9N17S network. The following thoughts come from this fact.

Table 8.II: Obtained lower bounds.

network	availability	solution from [61]	CG lower bound	gap
9N17S	99.9984 %	69.49 %	65.46 %	6.16 %
	99.9980 %	–	63.72 %	–
	99.9976 %	66.10 %	63.56 %	3.97 %
	99.9970 %	66.10 %	63.56 %	3.97 %
COST239	99.9940 %	–	53.58 %	–
	99.9920 %	–	53.58 %	–
	99.9900 %	–	53.58 %	–

Firstly, the bounds obtained by the CG algorithm are quite tight. Secondly, the quality of the integer solutions are not too far from optimality, although there is still some room for improvement. It is worth to mention that we also solved the linear relaxation of the p-cycle design problem without availability constraint, using the algorithm presented in Chapter 5, and the redundancy obtained for 9N17S was 63.56%, i.e, the same value as for 99.9976% and 99.9970% maximum availability. Also for COST239, the same value of redundancy, namely 53.58% was found. This means that these availability requirements does not affect the cost of the optimal LP solutions for these networks.

With the purpose of going further with our evaluation of the obtained lower bounds, we focused on finding further integer solutions by using all generated columns to obtain an ILP model and solving it with CPLEX solver. However, our attempt was not very successful as it is shown in Table 8.III. For 9N17S network, we could not find feasible solutions for the values of availability shown in 8.II. Thus, we show a result obtained for a smaller availability limit so that we can have some information about possible gaps. We end up with a solution having a quite large gap, much larger than those for the solutions from [61], which does not bring much into a conclusion. As for COST239, we again obtained a solution with very large optimality gap.

As final comments, it is clear that the method proposed in [61] appears to be much more successful in finding integer solutions for practical purposes by being able to provide good results in the presence of more rigorous availability requirements. As for the contribution of this study, we provided the first lower bounds on optimal solutions for

Table 8.III: Obtained integer solutions.

network	availability	integer solution	gap
9N17S	99.9960 %	40.68 %	28.00 %
COST239	99.9940 %	37.64 %	40.51 %

the problem. Our preliminary experiments give us an indication that these bounds are rather tight, whereas it may be possible that better integer solutions exist.

An immediate future work would be to derive additional columns from the solutions obtained in [61] so as to give more flexibility to CPLEX MIP solver during the search for possibly better solutions. Another future research direction is to either investigate possible ways of making the CG algorithm more scalable in order to approach larger instances as well as more stringent requirements, or develop effective heuristics which could help to validate the results from [61].

CHAPTER 9

CONCLUSION

Survivability plays a very important role in high speed transport networks, given the growth of Internet services and the enormous bandwidth capability of optical networks brought with WDM technology. Among the several survivability mechanisms, p-cycles appear to be very promising and attractive because they achieve a good trade-off between restoration time and capacity efficiency, as explained in Chapter 2.

In Chapter 3, we have seen that many research works have been carried out on p-cycle protection and its extensions since it was introduced in 1998. These studies cover different topics such as new solution approaches and design issues in p-cycle networks. Most of the existing solution methods suffer from either their strong heuristic nature or their lack of scalability. In the first case, no precise information about the quality of the solutions is provided and thus comparison with other such heuristic methods does not bring much light into concluding statements. In the other case, there is a severe limitation in providing results for relatively small network instances, mainly because these methods are usually based on the explicit enumeration of all cycles in the network.

The works presented in this thesis relied on solution methods which provided lower bounds on optimal solutions as well as accurate integer solutions. This was achieved by means of column generation based algorithms. In Chapter 4, we proposed a column generation formulation for FIPP p-cycles which overcame the existing methods and required reasonable computing times for solving the tested network instances. Indeed, we produced reductions of up to 37% in the cost of the solutions.

With an enhanced formulation for the pricing problem, we carried out an accurate comparison between shared protection schemes by using a unified column generation framework which produced near optimal solutions (Chapter 5). Then, we investigated the usage of network resources by undirected p-cycles and FIPP p-cycles in the presence of asymmetric traffic (Chapter 6). To pursue our study, we additionally presented column generation formulations for both directed p-cycle and FIPP p-cycle design problems. Our results showed that the use of asymmetric unidirectional links is very cost effective under asymmetric traffic scenarios and the difficulty implied for implementing asymmetry in transport networks may be worthwhile.

In Chapter 7, we conceived a hierarchical decomposition method which was able to approach much larger instances for the design of survivable networks with FIPP p-cycles. The new formulation of the pricing problem, together with an interesting decomposition strategy, turned out to be very effective in reducing computing times with respect to the previous column generation algorithm.

Finally, we proposed a column generation formulation for the availability-aware design of link-protecting p-cycle networks in order to provide the first lower bounds for the problem (Chapter 8). Our preliminary results indicated that the obtained bounds are quite tight when compared with existing integer solutions. Unfortunately, the algorithm requires further improvements since it was only capable of producing results for very small problem instances and for not so stringent availability limits.

Many further research directions can be followed to continue the works in this thesis. Firstly, let us suggest some improvements which could be done to the solution methods proposed here. Concerning the FIPP p-cycle design problem, one of the issues present in the results reported in Chapter 7 is the number of iterations of the column generation algorithm, which indicates a convergence problem. Factors such as the size of the master problem, too much degeneracy in the restricted master problem, and symmetry in the columns can affect convergence speed of the overall algorithm. To overcome this shortcoming, we could investigate stabilization techniques to improve convergence. More particularly, we could try to restrict dual variables oscillation, mainly in the first iterations, by using a good guess for the optimal values of the dual variables [28]. Good initial solutions obtained by an efficient heuristic could help with this task. Also, we could investigate valid inequalities to strengthen the formulation of the pricing problem thereby possibly accelerating the CG algorithm.

As soon as we will have obtained a faster column generation algorithm for solving the LP relaxation, we could invest some effort in developing more sophisticated methods for obtaining integer solutions, such as a branch-and-price scheme. For that, we should investigate proper branching rules, which appears to be one of the biggest challenges for the effectiveness of such schemes.

As for the work presented in Chapter 8, the most immediate improvement should tackle the scalability issue due to the size of the pricing problem. Namely, heuristic algorithms should be investigated in order to speed up the solution of the pricing problem. We could also search for inspiration in existing heuristic methods for the original problem. Furthermore, additional columns extracted from integer solutions obtained by the method proposed in [61] could help the MIP solver to hopefully find feasible and better integer solutions, and thus contribute to a better validation of these existing solutions.

Regarding other research directions, we could address further design issues in p-cycle networks. For instance, the cost of optical-electrical-optical (O/E/O) in DWDM networks are considered as one of the dominant costs in building optical networks. These costs are usually measured by the number of optical ports required to be installed at the physical nodes. Therefore, as future work, we could also focus on the optimization of FIPP p-cycles networks with respect to equipment installation costs instead of capacity planning.

Another important design issue is the restoration time. p-Cycles have been claimed very fast protection schemes because they provide fully pre-connected protection paths. However, it would be interesting to provide precise information about the restoration times implied by p-cycles and FIPP p-cycles in comparison with classical shared link and path protection, discussed in Chapter 5. Some studies in this direction have been carried out in [49, 91, 103], but none of them included FIPP p-cycles. This future study should consider to perform an accurate evaluation of the trade-off between capacity usage and restoration time present in optimal or quasi-optimal solutions for each protection scheme.

BIBLIOGRAPHY

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network flows: theory, algorithms, and applications*. Prentice Hall, 1993.

[2] A. Atamtürk and D. Rajan. Partition inequalities for capacitated survivable network design based on directed p-cycles. *Discrete Optimization*, 5:415–433, 2008.

[3] D. Baloukov, W.D. Grover, and A. Kodian. Toward jointly optimized design of failure-independent path-protecting p-cycle networks. *Journal of Optical Networking*, 7(1):62–79, January 2008.

[4] R. Bandhari. *Survivable Networks: Algorithms for Diverse Routing*. Kluwer Academic Plubishers, 1999.

[5] 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, May-June 1998.

[6] P. Batchelor, B. van Caenegem, B. Daino, P. Heinzmann, D.R. Hjelme, R. Inkret, H.A. Jäger, M. Joindot, A. Kuchar, E. Le Coquil, P. Leuthold, G. de Marchis, F. Matera, M. Mikac, H.P. Nolting, J. Späth, F. Tillerot, N. Wauters, and C. Weinert. Ultra high-capacity optical transmission networks: Final report of action COST 239. Technical report, Faculty of Electrical Engineering and Computing, University of Zagreb, 1999.

[7] N. Bhuiyan. Segment protection in WDM networks. Master's thesis, Departement of Computer Science and Software Engineering, Concordia Univeristy, Montréal, Canada, December 2008.

[8] G. Birkan, J. Kennington, E. Olinick, A. Ortynski, and G. Spiride. Making a case for using integer programming to design dwdm networks. *Optical Networks Magazine*, 4(6):107–120, 2003.

[9] A.K. Bjerring and B.St. Arnaud. A.k. bjerring and b. st. arnaud. optical internets and their role in future telecommunications systems. Technical report, Internal CA*net4 document, 2001.

[10] I.M. Bomze, M. Budinich, P.M. Pardalos, and M. Pelillo. The maximum clique problem. In *Handbook of Combinatorial Optimization*, pages 1–74. Kluwer Academic Publishers, 1999.

[11] E. Calle, J.L. Marzo, and A. Urra. Protection performance components in MPLS networks. *Computer Communications*, 27:1220–1228, 2004.

[12] P. Cholda and A. Jajszczyk. Reliability assessment of optical p-cycles. *IEEE/ACM Transactions on Networking (TON)*, 15(6):1579–1592, December 2007.

[13] T.Y. Chow and P.J. Lin. The ring grooming problem. *Networks*, 44:194–202, 2004.

[14] V. Chvatal. *Linear Programming*. Freeman, 1983.

[15] M. Clouqueur. *Availability of service in mesh-restorable transport networks*. PhD thesis, University of Alberta, 2004.

[16] M. Clouqueur and W.D. Grover. Availability analysis of span-restorable mesh networks. *IEEE Journal on Selected Areas in Communications*, 20(4):810–821, May 2002.

[17] M. Clouqueur and W.D. Grover. Availability analysis and enhanced availability design in p-cycle-based networks. *Photonic Network Communications*, 10(1): 55–71, July 2005.

[18] M. Clouqueur and W.D. Grover. Mesh-restorable networks with enhanced dual-failure restorability properties. *Photonic Network Communications*, 9(1):7–18, January 2005.

[19] A.L. Corcoran and R.L. Wainwright. A user-friendly workbench for order-based genetic algorithm research. In *Proceedings of ACM/SIGAPP Symposium on Applied Computing (SAC 93)*, Indianapolis, Indiana, 1993.

[20] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, 2nd edition, 2001.

[21] D. Crawford. Fiber optic cable dig-ups: Causes and cures. In *Network Reliability: A Report to the Nation - Compendium of Technical Papers*. National Engineering Consortium, Chicago, 1993.

[22] G. Dahl and M. Stoer. A cutting plane algorithm for multicommodity survivable network design problems. *INFORMS Journal on Computing*, 10(1):1–11, 1998.

[23] G. Dantzig. *Linear Programming and Extensions*. Princeton University Press, Princeton, 1963.

[24] G. Desaulniers, J. Desrosiers, and M. M. Solomon, editors. *Column Generation*, GERAD 25th Anniversary Series, 2005. Spring.

[25] J. Doucette. *Advances on Design and Analysis of Mesh-Restorable Networks*. PhD thesis, University of Alberta, Edmonton, AB, Canada, 2004.

[26] J. Doucette, D. He, W. Grover, and O. Yang. Algorithmic approaches for efficient enumeration of candidate p-cycles and capacitated p-cycle network design. In *Proceedings of the Fourth International Workshop on Design of Reliable Communication Networks (DRCN)*, pages 212–220, October 2003.

[27] J. Doucette, P. Giese, and W.D. Grover. Investigation of node protection strategies with node-encircling p-cycles. In *Proceeding of the International Workshop on the Design of Reliable Communication Networks (DRCN 2005)*, Ischia, Italy, October 2005.

[28] O. du Merle, D. Villeneuve, J. Desrosiers, and P. Hansen. Stabilized Column Generation. *Discrete Mathematics*, 194(1):229–248, 1999.

[29] A.F. Elrefaie. Multiwavelength survivable ring network architecture. In *Proceedings of the IEEE International Conference on Communications (ICC)*, pages 1245–1251, May 1993.

[30] D. Epstein. Finding the k shortest paths. *SIAM J. Computing*, 28(2):652–673, 1998.

[31] A.E. Eshoul and H.T. Mouftah. Survivability approaches using p-cycles in wdm mesh networks under static traffic. *IEEE/ACM Transactions on Networking (TON)*, 17(2):671–683, April 2009.

[32] M. Fischetti, J. J. Salazar Gonzalez, and P. Toth. A branch-and-cut algorithm for the symmetric generalized traveling salesman problem. *Operations Research*, 45 (3):378–394, May-June 1997.

[33] R.L. Freeman. *Fiber-Optic Systems for Telecommunications*. Wiley, New York, 2002. ch. 6 and 10.

[34] A. Fumagalli and L. Valcarenghi. IP restoration vs. WDM protection: is there an optimal choice? *IEEE Network*, 14(6):34–41, 2000.

[35] C. Ge, N. Bai, X. Sun, and M. Zhang. Iterative joint design approach for failure-independent path-protecting p-cycle networks. *Journal of Optical Networking*, 6 (12):1329–1339, November 2007.

[36] P. C. Gilmore and R. E. Gomory. A linear programming approach to the cutting stock problem. *Operations Research*, 9:849–859, 1961.

[37] P. C. Gilmore and R. E. Gomory. A linear programming approach to the cutting stock problem: Part ii. *Operations Research*, 11:863–888, 1963.

[38] W. D. Grover. *Mesh-Based Survivable Networks*. Prentice Hall, 2004.

[39] W. D. Grover and D. Stamatelakis. Cycle-oriented distributed preconfiguration: Ring-like speed with mesh-like capacity for self-planning network restoration. In

IEEE International Conference on Communications (ICC 1998), pages 537–543, June 1998.

[40] W.D. Grover. The protected working capacity envelope concept: An alternative paradigm for automated service provisioning. *IEEE Communications Magazine*, 42(1):62–69, 2004.

[41] W.D. Grover. www.ece.ualberta.ca/˜grover/book/.

[42] W.D. Grover and J.E.Doucette. Advances in optical network design with p-cycles: Joint optimization and pre-selection of candidate p-cycles. In *Proceedings of the IEEE/LEOS Summer Topical Metting on All-Optical Networking*, pages WA2–49 – WA2–50, Mont Tremblant, Canada, July 2002.

[43] W.D. Grover and A. Kodian. Failure-independent path-protection with p-cycles: Effcient, fast and simple protection for transparent optical networks. In *Proceedings of the 7th International Conference on Transparent Optical Networks (ICTON)*, volume 1, pages 363–369, July 2005.

[44] W.D. Grover and A. Sack. High availability survivable networks: When is reducing MTTR better than adding protection capacity? In *Proceedings of IEEE Workshop on Design of Reliable Communication Networks (DRCN)*, La Rochelle, France, October 2007.

[45] W.D. Grover and G. Shen. Extending the p-cycle concept to path-segment protection. In *IEEE International Conference on Communications (ICC 2003)*, pages 1314–1319, May 2003.

[46] W.D. Grover and D. Stamatelakis. Bridging the ring-mesh dichotomy with p-cycles. In *Proceedings of IEEE/VDE Workshop on Design of Reliable Communication Networks (DRCN)*, pages 92–104, Munich, Germany, April 2000.

[47] W.D. Grover, J. Doucette, A. Kodian, D. Leung, A. Sack, M. Clouqueur, and G. Shen. Design of survivable networks based on p-cycles. In P. Pardalos and

M. Resende, editors, *Handbook for Optimization in Telecommunications*, chapter 16, pages 391–434. Springer, 2006.

[48] H. Harai, M. Murata, and H. Miyahara. Performance analysis of wavelength assignment policies in all-optical networks with limited-range wavelength conversion. *IEEE Journal on Selected Areas in Communications*, 16(7):1051–1060, 1998.

[49] W. He and A.K. Somani. Comparison of protection mechanisms: Capacity efficiency and recovery time. In *IEEE International Conference on Communications (ICC)*, pages 2218–2223, June 2007.

[50] P. Ho and T.M. Hussein. A framework for service-guaranteed shared protection in WDM mesh networks. *IEEE Communications Magazine*, 40(2):97–103, February 2002.

[51] P.-H. Ho, J. Yapolcai, and T. Cinkler. Shared segment protection in mesh communication networks with bandwidth guaranteed tunnels. *IEEE/ACM Transactions on Networking*, 12(6):1105–1118, December 2004.

[52] P.H. Ho and H.T. Mouftah. SLSP: a new path protection scheme for the optical internet. In *Optical Fiber Communications (OFC)*, volume 2, pages TuO1–1–TuO1–3, 2001.

[53] R. Hülsermann, S. Bodamer, M. Barry, A. Betker, C. Gauger, M. Jäger, M. Köhn, and J. Späth. A set of typical transport network scenarios for network modelling. Technical report, ITG-Fachtagung Photonische Netze, Leipzig, 2004.

[54] R.R. Iraschko, M. MacGregor, and W.D. Grover. Optimal placement for path restoration in stm or atm mesh survivable networks. *IEEE/ACM Transactions on Networking*, 6(3):325–336, June 1998.

[55] ITU. Spectral grids for WDM applications: DWDM frequency grid. Recommendation G.694.1, June 2002.

[56] ITU. Spectral grids for WDM applications: CWDM frequency grid. Recommendation G.694.2, December 2003.

[57] B. Jaumard, C. Rocha, D. Baloukov, and W. D. Grover. A column generation approach for design of networks using path-protecting p-cycles. In *Proceedings of the Sixth International Workshop on the Design of Reliable Communication Networks (DRCN)*, October 2007.

[58] W. John, S. Tafvelin, and T. Olovsson. Trends and differences in connection-behavior within classes of internet backbone traffic. In *Proceedings of the 9th International Conference on Passive and Active Measurement*, number 4879 in Lectures Notes in Computer Science, pages 192–201, Germany, April 2008. Springer-Verlag.

[59] D.B. Johnson. Finding all the elementary circuits of a directed graph. *SIAM Journal on Computing*, 4(1):77–84, 1975.

[60] J.L. Kennington, E.V. Olinick, and G. Spiride. Basic mathematical programming models for capacity allocation in mesh-based survivable networks. *Omega*, 35: 629–644, 2007.

[61] M.S. Kiaei, A. Ranjbar, C. Rocha, B. Jaumard, and C. Assi. Improved availability models for p-cycle-based network design. In *Proceedings of IEEE International Workshop on the Design of Reliable Communication Networks*, October 2009. (to appear).

[62] S. Kini, M. Kodialam, T.V. Laksham, S. Sengupta, and C. Villamizar. Shared backup label switched path restoration. *IEEE/ACM Transactions on Networking*, May 2001. IETF Internet Draft, work in progress.

[63] A. Kodian. *Advances in p-Cycle Network Design*. PhD thesis, University of Alberta, Edmonton, Alberta, Canada, 2006.

[64] A. Kodian and W.D. Grover. Failure-independent path-protecting p-cycles efficient and simple fully preconnected optical-path protection. *Journal of Lightwave Technology*, 23(10):3241–3259, October 2005.

[65] A. Kodian, W.D. Grover, J. Slevinsky, and D. Moore. Ring-mining to p-cycles as a target architecture: Riding demand growth into network efficiency. In *19th Annual National Fiber Optics Engineers Conference (NFOEC 2003)*, pages 1543–1552, September 2003.

[66] A. Kodian, W.D. Grover, and J. Doucette. A disjoint route sets approach to design of failure-independent path-protection p-cycle networks. In *Proceedings of the International Workshop on the Design of Reliable Communication Networks (DRCN)*, October 2005. 8pp.

[67] A. Kodian, A. Sack, and W.D. Grover. The threshold hop-limit effect in p-cycles: Comparing hop- and circumference-limited design. *Optical Switching and Networking*, 2:72–85, September 2005.

[68] A. M. C. A. Koster, A. Zymolka, M. Jager, and R. Hülsermann. Demand-wise shared protection scheme for meshed optical networks. *Journal of Network and Systems Management*, 13(1):35–55, March 2005.

[69] G. Laporte and I.R. Martín. Locating a cycle in a transportation or a telecommunications network. *Networks*, 50(1):92–108, 2007.

[70] D. Leung, S. Arakawa, M. Murata, and W.D. Grover. Re-optimization strategies to maximize traffic-carrying readiness in WDM survivable mesh networks. In *Proceedings of Optical Fiber Communication Conference (OFC)*, volume 3, March 2005.

[71] C. Liu and L. Ruan. Finding good candidate cycles for efficient p-cycle network design. In *Proceedings of the 13th International Conference on Computer Communications and Networks (ICCCN)*, pages 321–326, October 2004.

[72] C. Liu and L. Ruan. Dynamic provisioning of survivable services using path-segment protecting p-cycles in WDM networks. In *Proceedings of the 15th International Conference on Computer Communications and Networks (ICCCN 2006)*, pages 275–280, October 2006.

[73] K. Lo, D. Habibi, Q.V. Phung, A. Rassau, and H.N. Nguyen. Efficient p-cycles design by heuristic p-cycle selection and refinement for survivable wdm mesh networks. In *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, 2008.

[74] I. Loiseau, A. Ceselli, N. Maculan, and M. Salani. Génération de colonnes en programmation linéaire en nombres entiers. In V. Paschos, editor, *Optimisation combinatoire 1 : concepts fondamentaux*, SÃl'rie Informatique et SystÃĺmes d'Information, pages 237–262. Hermes Science publications - Lavoisier, 2005.

[75] M. E. Lübbecke and J. Desrosiers. Selected topics in column generation. *Operations Research*, 53(6):1007–1023, 2005.

[76] N. Maculan, M. M. Passini, J. A. M. Brito, and A. Lisser. Column generation method for network design. In M. Gendreau and P. Marcotte, editors, *Transportation and Netowrk Analysis: Current Trends*, pages 167–179. Kluwer Academic Publishers, 2002.

[77] N. Maculan, M. M. Passini, J. A. M. Brito, and I. Loiseau. Column generation in integer linear programming. *RAIRO-Operations Research*, 37:67–83, 2003.

[78] M.W. Maeda. Management and control of transparent optical networks. *IEEE Journal on Selected Areas in Communications*, 16(7):1005–1023, September 1998.

[79] S. De Maesschalck, M. Pickavet, D. Colle, and P. Demeester. Influence of the IP traffic asymmetry on the cost of the optical network layer. *Photonic Network Communications*, 4(2):133–150, 2002.

[80] S. De Maesschalck, D. Colle, I. Lievens, M. Pickavet, P. Demeester, C. Mauz, M. Jaeger, R. Inkret, B. Mikac, and J. Derkacz. Pan-european optical transport networks: An availability-based comparison. *Photonic Network Communications*, 5(3):203–225, May 2003.

[81] G. Maier, A. Pattavina, S. De Patre, and M. Martinelli. Optical network survivability: Protection techniques in the WDM layer. *Photonic Network Communications*, 41(3/4):251–269, July 2002.

[82] C. Mauz. p-cycle protection in wavelength routed networks. In *Proceedings of the Working Conference on Optical Network Design and Modelling (ONDM)*, Budapest, Hungary, February 2003.

[83] C. Mauz. Unified ILP formulation of protection in mesh networks. In *Proceedings of the International Conference on Telecommunications/Workshop on All-Optical Networks*, pages 737–741, Zagren, Croatia, June 2003.

[84] G. Mohan and C.S.R. Murthy. Lightpath restoration in WDM optical networks. *IEEE Network Magazine*, 6(14):24–32, November/December 2000.

[85] D.S. Mukherjee, C. Assi, and A. Agarwal. An alternative approach for enhanced availability analysis and design methods in p-cycle based networks. *IEEE Journal on Selected Areas in Communication*, 24:23–34, 2006.

[86] P. Neusy and R. Habel. Availability analysis of optical shared protection rings for long-haul networks. In *Optical Fiber Communications (OFC)*, volume 1, pages 176–179, February 1999.

[87] T.F. Noronha and C.C. Ribeiro. Routing and wavelength assignment by partition coloring. *European Journal of Operational Research*, 171(3):797–810, 2006.

[88] D.P. Onguetou and W.D. Grover. Approaches to p-cycle network design with controlled optical path lentghs n the restored network state. *Journal of Optical Networking*, 7(7):673–691, July 2008.

[89] D.P. Onguetou and W.D. Grover. Solution of a 200-node p-cycle network design problem with GA-based pre-selection of candidate structures. In *Proceedings of the IEEE International Conference on Communications (ICC)*, Dresden, Germany, June 2009.

[90] S. Orlowski, M. Pióro, A. Tomaszewski, and R. Wessäly. SNDlib 1.0–Survivable Network Design Library. In *Proceedings of the Third International Network Optimization Conference (INOC)*, Spa, Belgium, April 2007. http://sndlib.zib.de.

[91] Y. Ouyang and Q. Zeng. Connection provisioning with guaranteed recovery time in survivable WDM optical networks. *Photonic Network Communications*, 12: 309–319, 2006.

[92] M.W. Padberg. On the facial structure of set packing polyhedra. *Mathematical Programming*, 5:199–215, 1973.

[93] J.P. Madhavarapu P.K. Gummadi and C.S.R. Murthy. A segment backup scheme for dependable real time communication in multihop networks. In *Proceedings of 15th Workshop on Parallel and Distributed Processing*, pages 678–684, 2000.

[94] D. Rajan and A. Atamtürk. Survivable network design: Routing of flows and slacks. In G. Anandalingam and S. raghavan, editors, *Telecommunications Network Design and Management*, chapter 4, pages 65–81. Kluwer Academic Publishers, 2002.

[95] R. Ramaswami and K.N. Sivarajan. *Optical networks: A practical perspective*. Morgan Kaufmann, 2nd edition, 2002.

[96] A. Ranjbar and C. Assi. Availability-aware design in FIPP p-cycles protected mesh networks. In *Proceedings of the International Conference on Optical Network Design and Modeling (ONDM 2008)*, Vilanova i la Geltrú, Spain, March 2008.

[97] Reuters. Chinese web users lose 10,000 domain names in quakes. `http://www.reuters.com/article/technologyNews/idUSSHA15067820070105`, January 2007.

[98] C. Rocha and B. Jaumard. Revisiting p-cycles / FIPP p-cycles vs. shared link / path protection. In *17th International Conference on Computer Communications and Networks (ICCCN)*, St. Thomas, USA, 2008.

[99] C. Rocha and B. Jaumard. A hierarchical decomposition method for efficient computation of FIPP p-cycles. *Submitted to publication*, 2009.

[100] C. Rocha, B. Jaumard, and P.-E. Bougué. Directed vs. undirected p-cycles and FIPP p-cycles. In *Proceedings of the International Network Optimization Conference (INOC)*, Pisa, Italy, April 2009.

[101] C. Rocha, B. Jaumard, and P.-E. Bougué. Asymmetry issues in p-cycle and FIPP p-cycle protection schemes. *Submitted to publication*, 2009.

[102] A. Sack and W.D. Grover. Hamiltonian p-cycles for fiber-level protection in homogeneous and semi-homogeneous optical networks. *IEEE Network*, 18(2):49–56, March-April 2004.

[103] L. Sahasrabuddhe, S. Ramamurthy, and B. Mukherjee. Fault management in IP-over-WDM networks: WDM protection versus IP restoration. *IEEE Journal of Selected Areas in Communications*, 20(1):21–33, January 2002.

[104] D. A. Schupke, C. G. Gruber, and A. Autenrieth. Optimal configuration of p-cycles in WDM networks. In *Proceedings of the IEEE International Conference on Communications (ICC)*, volume 5, pages 2761–2765, May 2002.

[105] D.A. Schupke. Link and path asymmetry issues in IP over WDM transport networks. In *Fifth Working Conference on Optical Network Design and Modelling (ONDM)*, Vienna, Austria, February 2001.

[106] D.A. Schupke. Multiple failure survivability in wdm networks with p-cycles. In *Proceedings of the International Symposium on Circuits and Systems (ISCAS)*, volume 3, pages III–866–III–869, May 2003.

[107] D.A. Schupke. The tradeoff between the number of deployed p-cycles and survivability to dual fiber duct failures. In *Proceedings of the IEEE International Conference on Communications (ICC)*, volume 2, pages 1428–1432, Anchorage, AK, May 2003.

[108] D.A. Schupke. An ILP for optimal p-cycle selection without cycle enumeration. In *Proceedings of the 8th Working Conference on Optical Network Design and Modelling (ONDM)*, Ghent, Belgium, February 2004.

[109] D.A. Schupke. Analysis of p-cycle capacity in WDM networks. *Photonic Network Communications*, 12(1):41–51, July 2006.

[110] D.A. Schupke, M.C. Scheffel, and W.D. Grover. Configuration of p-cycles in wdm networks with partial wavelength conversion. *Photonic Network Communications*, 6(3):239–252, November 2003.

[111] D.A. Schupke, W.D. Grover, and M. Clouqueur. Strategies for enhanced dual failure retorability with static or reconfigurable p-cycle networks. In *Proceedings of the IEEE International Conference on Communications (ICC)*, pages 20–24, Paris, France, 2004.

[112] S. Sebbah and B. Jaumard. Efficient and scalable design of protected working capacity envelope. In *NETWORKS*, 2008.

[113] S. Sebbah and B. Jaumard. Survivable WDM networks design with non-simple p-cycle-based PWCE. In *GLOBECOM*, 2008.

[114] S. Sebbah and B. Jaumard. p-Cycle based dual failure recovery in WDM mesh networks. In *Proceedings of IEEE International Conference on Optical Network Design and Modeling (ONDM)*, February 2009.

[115] S. Sengupta and R. Ramamurthy. Capacity efficient distributed routing of mesh-restored lightpaths in optical networks. In *Proceedings of IEEE Global Telecommunications Conference (GlobeCom 2001)*, pages 2129–2133, San Antonio, TX, November 2001.

[116] G. Shen and W. D. Grover. Performance of protected working capacity envelopes based on p-ycles: Fast, simple, and scalable dynamic service provisioning of survivable services. In *Proceedings of APOC*, volume 5626, pages 519–533, 2004.

[117] G. Shen and W. D. Grover. Design and performance of protected working capacity envelopes based on *p*-cycles for dynamic provisioning of survivable services. *Journal of Optical Networking*, 4:361–390, July 2005.

[118] G. Shen and W.D. Grover. Extending the p-cycle concept to path segment protection for span and node failure recovery. *IEEE Journal on Selected Areas in Communications*, 21(8):1306–1319, 2003. ISSN 0733-8716.

[119] M. Sivakumar, R.K. Shenai, and K.M. Sivalingam. A survey of survivability techniques for optical WDM networks. In *Emerging Optical Network Technologies*, chapter 13. Springer, 2006.

[120] K.M. Sivalingam and S. Subramaniam. *Emerging Optical Network Technologies : Architectures, Protocols, and Performance*. Springer, New York, 2005.

[121] J. Smith, A. Schaefer, and J. Yen. A stochastic integer programming approach to solve a synchronous optical network ring design problem. *Networks*, 44(1): 12–26, 2004.

[122] D. Stamatelakis and W. D. Grover. Theoretical underpinnings for the efficiency of restorable networks using pre-configured cycles p-cycles. *IEEE Transactions on Communications*, 48(8):1262–1265, 2000.

[123] D. Stamatelakis and W.D. Grover. Network restorability design using pre-configured trees, cycles and mixtures of pattern types. Technical Report TR-1999-05, TRLabs, Edmonton, AB, Canada, October 2000.

[124] D. Stamatelakis and W.D. Grover. IP layer restoration and network planning based on virtual protection cycles. *IEEE Journal on Selected Areas in Communications*, 18(10):1938–1949, October 2000.

[125] T.E. Stern and K. Bala. *Multiwavelength Optical Networks: A Layered Approach*. Prentice Hall, New Jersey, 1999.

[126] T.E. Stern, G. Ellinas, and K. Bala. *Multiwavelength Optical Networks: Architectures, design, and Control*. Cambridge University Press, second edition, 2008.

[127] T. Stidsen and T. Thomadsen. Joint optimization of working and p-cycle protection capacity. Technical report, Technical Univ. of Denmark, 2004.

[128] T. Stidsen, B. Petersen, K.B. Rasmussen, S. Spoorendonk, M. Zachariasen, F. Rambach, and M. Kiese. Optimal routing with single backup path protection. In *International Network Optimization Conference (INOC)*, 2007.

[129] J. Tapolcai and P. Ho. Linear formulation for segment shared protection. In *Proceedings of the SPIE OptiComm 2003*, pages 49–58. SPIE, oct 2003. doi: 10.1117/12.533176.

[130] T. Thomadsen and T. Stidsen. The quadratic selective travelling salesman problem. Technical report, Technical Univ. of Denmark, 2003.

[131] K. Thompson, G.J. Miller, and R. Wilder. Wide-area Internet traffic patterns and characteristics. *IEEE/ACM Transactions on Networking*, pages 10–23, November 1997.

[132] M. To and P. Neusy. Unvailability analysis of long-haul networks. *IEEE Journal of Selected Areas in Communication*, 12(1):100–109, 1994.

[133] J. Wang, L. Sahasrabuddhe, and B. Mukherjee. Path vs. subpath vs. link restoration for fault management in IP-over-WDM networks: performance comparisons using GMPLS control signaling. *IEEE Communications Magazine*, 40:80–82, November 2002.

[134] B. Wu, K.L. Yeung, and S. Xu. A new ILP-based p-cycle construction algorithm without candidate cycle enumeration. In *Proceedings of the IEEE International Conference on Communications (ICC)*, June 2007.

[135] B. Wu, K.L. Yeung, and S. Xu. ILP formulation for p-cycle construction based on flow conservation. In *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, November 2007.

[136] T.H. Wu. *Fiber Network Service Survavibility*. Artech House Publishers, 1992.

[137] F. Zhang and W.-D. Zhong. A novel path-protecting p-cycle heuristic algorithm. In *Proceeding of the International Conference on Transparent Optical Networks (ICTON)*, volume 3, pages 203–206, Nottingham, UK, June 2006.

[138] H. Zhang and O. Yang. Finding protection cycles in DWDM networks. In *Proceedings of IEEE International Conference on Communications (ICC 2002)*, volume 5, pages 2756–2760, 2002.

[139] J. Zhang and B. Mukherjee. A review of fault management in WDM mesh networks: basic concepts and research challenges. *IEEE Network*, 18:41–48, Mach/April 2004.

[140] Z. Zhang, W.D. Zhong, and B. Mukherjee. A heuristic algorithm for p-cycles configuration in WDM optical networks. In *Proceedings of the Opto-Electronics and Communications Conference (OECC)*, pages 685–686, Shanghai, China, October 2003.

[141] Z. Zhang, W. Zhong, and S. B. Bose. Dynamically survivable WDM network design with p-cycle-based PWCE. *IEEE Communications Letters*, 9(8):756–758, August 2005.

[142] Z.R. Zhang, W.D Zhong, and B. Mukherjee. A heuristic method for design of survivable WDM networks with p-cycles. *IEEE Communications Letters*, 8(7): 467–469, July 2004.