

**University of Alberta**

Studies in Failure Independent Path-Protecting  $p$ -Cycle Network Design

by

Dimitri Baloukov

A thesis submitted to the Faculty of Graduate Studies and Research  
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Electrical and Computer Engineering

© Dimitri Baloukov  
Fall 2009  
Edmonton, Alberta

Permission is hereby granted to the University of Alberta Libraries to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only. Where the thesis is converted to, or otherwise made available in digital form, the University of Alberta will advise potential users of the thesis of these terms.

The author reserves all other publication and other rights in association with the copyright in the thesis and, except as herein before provided, neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatsoever without the author's prior written permission.

# Examining Committee

---

Dr. Wayne D. Grover (Supervisor)

- Department of Electrical and Computer Engineering

Dr. Raymond DeCorby (Supervisory Committee Member)

- Department of Electrical and Computer Engineering

Dr. Jose Nelson Amaral, (External Examiner)

- Department of Computing Science

# Abstract

---

Failure Independent Path-Protecting (FIPP)  $p$ -Cycles is a recently proposed protection architecture for transport networks that extends the properties of mesh-like efficiency and ring-like speed of span-protecting  $p$ -cycles to path protection. FIPP  $p$ -cycles provide shared end-to-end protection to working paths and exhibit properties of pre-connection, end-node activation and failure independence. In this thesis we advance the state of the art in FIPP  $p$ -cycle networking. We first introduce two new methods for FIPP  $p$ -cycle network design: FIPP column generation (CG) and FIPP iterative heuristic (IH). This is followed by the introduction of a new method for joint capacity placement design called FIPP disjoint route set (DRS) joint capacity placement (JCP) which is followed by an in-depth investigation on the effects of jointness in FIPP  $p$ -cycle designs. Next we introduce a series of comparative case studies involving several pre-connected network survivability architectures in the context of transparent optical networking. These studies include topics of single, dual and node failure restorability, minimum wavelength assignment and transparent reach analysis. The final contribution of this thesis is the investigation of the capital expenditure associated with the implementation of FIPP  $p$ -cycle designs using optical transport networking equipment as described in the NOBEL cost model. A new method called FIPP maximize unit path straddlers (MUPS) is introduced as part of this final study in order to utilize the property of same wavelength protection. This new approach is motivated by opportunities for cost reduction discovered in the initial costing exercise of the NOBEL cost model investigation.

# Acknowledgements

---

First, I would like to thank my supervisor Dr. Wayne D. Grover for his guidance, motivation, the opportunity to work as part of the Network Systems group at TRILabs as well as for financial support. You provided an interesting point of view and insight into the world of academic research as well as the field of engineering in general. Thank you for encouraging me to generate works of high integrity and quality and exemplifying these traits in your own projects.

I would also like to thank the current and past staff of TRILabs for making the work atmosphere in the office a pleasant one. Thank you Linda Richens, Rhoda Hayes, Jo-Ann Blumhagen, Luke Chong, Dave Clegg, Faith Goller and Tara Heron. I appreciate your help and support over the course of my graduate studies.

I would like to thank Dr. Matthieu Clouqueur and Dr. Dominic Schupke of Nokia Siemens Networks for their contributions to the HAVANA Project and for giving our group an opportunity to work on a challenging and fruitful project.

I would also like to thank Dr. Brigitte Jaumard from Concordia University and Caroline Rocha from Universite de Montreal for their expert contributions in the field of operations research and column generation.

A huge thank you to Aden Grue, Brian Forst and Diane Onguetou for their contributions to the work presented in this thesis. I always appreciated that I could bounce ideas off of you, be it work related or not.

Thank you to Dr. Adil Kodian, Dr. John Doucette, Dr. Gangxiang Shen and Anthony Sack for your help and guidance in the initial stages of my graduate studies.

I would also like to thank my committee members Dr. Jose Nelson Amaral and Dr. Ray DeCorby for your helpful comments and advice and Dr. Vincent Gaudet for volunteering his time to chair my thesis defense.

A very special thank you to my family for their encouragement and support. I would like to thank my brother Alex for proofreading this thesis for me. And a special thank you to Adam Elliot and Farren Hawkes for seeing me through to the end. I knew I could count on you being there when I needed it.

Dimitri Baloukov

September 22, 2009

# Table of Contents

---

<b>Chapter 1:</b>	<b>Introduction</b> . . . . .	<b>1</b>
1.1	Introduction. . . . .	1
1.2	Failure Independent Path-Protecting p-Cycles . . . . .	3
1.3	Thesis Outline and Highlights . . . . .	4
<b>Chapter 2:</b>	<b>Background</b> . . . . .	<b>9</b>
2.1	Introduction. . . . .	9
2.2	The Transport Network. . . . .	9
2.3	Transport Network Hierarchy . . . . .	10
2.4	The Network Model . . . . .	11
2.5	Virtual Logical Networks . . . . .	14
2.6	Demand. . . . .	16
2.7	Network Equipment . . . . .	17
	2.7.1 ADM/OADM . . . . .	17
	2.7.2 OXC . . . . .	18
	2.7.3 Optical Amplifiers and Regenerators . . . . .	21
2.8	Transmission Technology . . . . .	22
	2.8.1 SONET/SDH . . . . .	22
	2.8.2 WDM/DWDM/CWDM . . . . .	24
2.9	Network Survivability . . . . .	25
	2.9.1 Protection and Restoration. . . . .	26
	2.9.2 Redundancy. . . . .	27
2.10	Survivability at the Physical Layer . . . . .	29
	2.10.1 Automatic Protection Switching. . . . .	30
	2.10.2 Self-Healing Rings . . . . .	31
2.11	Survivability at the Logical Layer . . . . .	35
	2.11.1 Span Restoration . . . . .	35
	2.11.2 Path Restoration . . . . .	36
	2.11.3 Shared Backup Path Protection . . . . .	38
	2.11.4 Demand-Wise Shared Protection . . . . .	39
	2.11.5 Pre-Cross-Connected Trails . . . . .	40
	2.11.6 p-Cycles . . . . .	42
	2.11.7 FIPP p-Cycles . . . . .	44
	2.11.8 GPP p-Cycles. . . . .	44
	2.11.9 Flow p-Cycles . . . . .	44
2.12	Optimization and Linear Programming . . . . .	45
	2.12.1 Set Theory and Notation . . . . .	46
	2.12.2 Linear and Integer Linear Programing . . . . .	47
	2.12.3 Formulation of an LP/ILP model. . . . .	49
	2.12.4 Solving LP and ILP Problems. . . . .	51
	2.12.5 The Simplex Method . . . . .	53
	2.12.6 Duality. . . . .	59

2.12.7	Column Generation	61
2.13	Algorithms	64
2.13.1	Dijkstra's Algorithm	64
2.13.2	Cycle Generation - Depth First Search	66
2.14	Summary	67
<b>Chapter 3:</b>	<b>FIPP p-Cycles</b>	<b>69</b>
3.1	Introduction	69
3.2	Background	69
3.3	Working Path/Cycle Relationships	71
3.3.1	On-Cycle Relationship	71
3.3.2	Straddling Relationship	72
3.3.3	z-Case	73
3.4	Configurations	74
3.5	Path-Protecting p-Cycle Classes	75
3.5.1	Pure FIPP	76
3.5.2	FIPP with z-Cases	76
3.5.3	Non-DRS FIPP	77
3.5.4	Generalized Path-Protecting p-Cycles	78
3.6	Prior Literature	80
3.7	Summary	82
<b>Chapter 4:</b>	<b>Column Generation FIPP p-Cycle Design</b>	<b>83</b>
4.1	Introduction	83
4.2	Motivation for Column Generation	83
4.3	FIPP p-Cycle Models	85
4.4	FIPP p-Cycle Column Generation Method	86
4.4.1	FIPP p-Cycle Column Generation ILP	87
4.5	FIPP p-Cycle Iterative Heuristic Method	92
4.5.1	FIPP p-Cycle Iterative Heuristic ILP SubProblem	93
4.6	FIPP p-Cycle DRS method	94
4.6.1	FIPP p-Cycle DRS ILP	96
4.7	Solving the CG model	98
4.8	Experimental Setup	99
4.9	Results	100
4.10	Conclusion	105
<b>Chapter 5:</b>	<b>Joint FIPP p-Cycle Design</b>	<b>107</b>
5.1	Introduction	107
5.2	Joint Optimization of Working and Spare Capacity	107
5.3	FIPP p-Cycle DRS JCP Method	109
5.3.1	FIPP p-Cycle DRS JCP ILP	112
5.4	Experimental Setup	116
5.5	Results and Discussion	118
5.5.1	Basic Results	118

5.5.2	Varying the Number of Eligible Working Routes per Relation with DRS propagation	120
5.5.3	Varying the Type and Number of DRSs per Demand . . .	123
5.5.4	Varying the Eligible Route Set Inclusion Threshold . . . . .	126
5.5.5	Visual Inspection of Joint FIPP p-cycle Solutions . . . . .	131
5.6	Conclusion . . . . .	134
<b>Chapter 6:</b>	<b>FIPP p-Cycle Architecture Investigations . . . . .</b>	<b>137</b>
6.1	Introduction . . . . .	137
6.2	Background . . . . .	138
6.3	Minimum Capacity Designs . . . . .	139
6.3.1	Equipment Capability Assumptions and the z-Case . . . . .	140
6.3.2	Overprotection and Actual Protection Assignment . . . . .	142
6.3.3	Common Test Network and Demand Data . . . . .	144
6.3.4	Experimental Setup . . . . .	145
6.3.5	Results . . . . .	146
6.4	Dual-Failure Restorability . . . . .	150
6.4.1	Method . . . . .	151
6.4.2	Results . . . . .	152
6.5	Minimum Wavelength Assignment . . . . .	155
6.5.1	Minimum Wavelength Assignment ILP Model . . . . .	156
6.5.2	Reaching a 20 Wavelength Solution . . . . .	162
6.5.3	Joint DRS Model for FIPP p-Cycles . . . . .	164
6.6	Reach Limit Analysis . . . . .	167
6.6.1	Analysis of a Minimum Capacity Solution . . . . .	168
6.6.2	Analysis of a Minimum Capacity Under 20 Wavelengths Solution	169
6.6.3	Reach Limit Feasible Solution . . . . .	170
6.6.4	Overall Reach Limit Results . . . . .	172
6.7	Node Failure Restorability . . . . .	173
6.7.1	At-a-Glance Analysis . . . . .	174
6.7.2	Node Failures and On-Cycle Relationships . . . . .	174
6.7.3	Node Failures and Straddling Relationships . . . . .	176
6.7.4	Mutual Capacity Problem . . . . .	178
6.7.5	Single Node Failures in Single Span Failure Restorable Minimum Capacity Designs	179
6.7.6	First Come First Serve Method . . . . .	180
6.7.7	Optimal Conflict Resolution Method . . . . .	180
6.7.8	Results . . . . .	182
6.8	100% Node Failure Restorable Designs . . . . .	184
6.8.1	Experimental Setup and Assumptions . . . . .	185
6.8.2	Results . . . . .	186
6.8.3	Overall Node Failure Restorability Results . . . . .	188
6.9	Conclusion . . . . .	189

<b>Chapter 7:</b>	<b>FIPP <math>p</math>-Cycle Cost Optimization . . . . .</b>	<b>193</b>
7.1	Introduction . . . . .	193
7.2	NOBEL Cost Model . . . . .	194
7.2.1	Nodal Equipment . . . . .	196
7.2.2	Span Equipment . . . . .	198
7.2.3	Transmission Equipment . . . . .	200
7.2.4	Generalized Node Model . . . . .	200
7.3	FIPP $p$ -Cycle Implementation . . . . .	201
7.3.1	On-cycle Implementation . . . . .	203
7.3.2	Straddler Implementation . . . . .	203
7.4	New Minimum Hops 3k Limited Non-z-Case DRS Solutions. . .	205
7.5	Minimum Capacity Results . . . . .	207
7.6	Initial Costing Exercise Results . . . . .	208
7.7	New Approach: Maximize Same Wavelength Protection . . . . .	212
7.8	FIPP Maximum Unit Path Straddler Iterative Heuristic. . . . .	213
7.8.1	Working Route Generation . . . . .	215
7.8.2	Generate Minimum Hops Straddling Route ILP . . . . .	216
7.8.3	Generate Minimum Hops On-Cycle Route ILP . . . . .	219
7.8.4	Generate Maximum Unit Protection Straddler Configuration ILP . . . . .	221
7.9	Minimum WL Assignment for Unit Path Straddler Solutions . . .	227
7.10	Experimental Setup . . . . .	228
7.11	Results . . . . .	229
7.12	Conclusion. . . . .	234
<b>Chapter 8:</b>	<b>Conclusion . . . . .</b>	<b>235</b>
8.1	Main Contributions . . . . .	235
8.2	Publications . . . . .	238
8.3	Technical Reports/Presentations . . . . .	239
8.4	Topics for Further Research . . . . .	240
<b>Chapter 9:</b>	<b>References . . . . .</b>	<b>243</b>
<b>Chapter 10:</b>	<b>Appendix . . . . .</b>	<b>251</b>
10.1	COST 239 [11] . . . . .	251
10.1.1	Topology . . . . .	251
10.1.2	Demand Pattern 1 . . . . .	252
10.1.3	Demand Pattern 2 . . . . .	252
10.2	Germany Network [10] . . . . .	252
10.2.1	Topology . . . . .	252
10.2.2	Demand Pattern. . . . .	253
10.3	15n30s1 Network Family [70]. . . . .	254
10.3.1	Topology . . . . .	254
10.3.2	Family Members' Span Lists. . . . .	256
10.3.3	Demand Pattern. . . . .	256

# List of Tables

---

TABLE 2.1.	Data rates for American and European Plesiochronous Digital Hierarchies. . . . .	23
TABLE 2.2.	Standard SONET/SDH data rates. . . . .	24
TABLE 4.1.	Results obtained by the FIPP CG algorithm. . . . .	102
TABLE 4.2.	FIPP DRS results. . . . .	103
TABLE 4.3.	FIPP IH results. . . . .	104
TABLE 5.1.	Initial results for FIPP, p-cycle and SBPP JCP methods using 5 eligible working route options per demand. The table also contains the results for the FIPP SCP solution. . . . .	119
TABLE 5.2.	Number of working route choices per demand that fall under the threshold of the first, second and third trials. The number of working route choices per demand for a threshold of 200% is also given. . . . .	129
TABLE 5.3.	Working flow distribution for the route inclusion threshold for trial 1, 2 and 3. Each column corresponds to an iteration where a particular threshold was set. For each iteration, the types of working routes used as well as the number of demand units routed among them is shown. . . . .	130
TABLE 5.4.	Design costs and the number of cycles used in sample SCP and JCP solutions. . . . .	133
TABLE 6.1.	FIPP p-cycle single-failure restorable designs: Minimum capacity test set trials. z-Case not removed. . . . .	147
TABLE 6.2.	FIPP p-cycle single-failure restorable designs: Minimum cost test set trials. z-Case not removed. . . . .	147
TABLE 6.3.	FIPP p-cycle single-failure restorable designs: Minimum capacity test set trials. z-Case removed. . . . .	148
TABLE 6.4.	FIPP p-cycle single-failure restorable designs: Minimum cost test set trials. z-Case removed. . . . .	149
TABLE 6.5.	Total capacities of the architectures under study. . . . .	150
TABLE 6.6.	Minimum capacity non-z-case solution R2 analysis without spare capacity stub release. . . . .	153
TABLE 6.7.	Minimum capacity non-z-case solution R2 analysis with spare capacity stub release. . . . .	153
TABLE 6.8.	Dual-failure restorability values the minimum capacity solutions for each architecture under study. . . . .	155

TABLE 6.9.	FIPP <i>p</i> -cycle wavelength assignment results based on minimum capacity non- <i>z</i> -case solutions from Section 6.3. . . . .	159
TABLE 6.10.	New FIPP <i>p</i> -cycle wavelength assignment results. . . . .	161
TABLE 6.11.	FIPP <i>p</i> -cycle 100% single-failure restorable JCP solution. Non- <i>z</i> -case, minimum capacity. . . . .	166
TABLE 6.12.	Minimum wavelength assignment values the minimum capacity solutions for each architecture under study. . . . .	166
TABLE 6.13.	FIPP <i>p</i> -cycle 100% single-failure restorable design: Minimum capacity, non- <i>z</i> -case, transparent reach feasible solution. . . . .	171
TABLE 6.14.	Total capacities of reach feasible solutions for each architecture under study. . . . .	173
TABLE 6.15.	Minimum capacity single span failure solutions: Comparison of results obtained using the FCFS and the Optimal Conflict Resolution method. . . . .	184
TABLE 6.16.	FIPP <i>p</i> -Cycle Single Node Failure Restorable Designs: Minimum Capacity Test Set. Nodal <i>z</i> -cases removed. . . . .	187
TABLE 6.17.	Minimum capacity single node failure restorable solutions: Node failure restoration using the Optimal Conflict Resolution method. . . . .	188
TABLE 6.18.	Total capacities of 100% node restorable minimum capacity solutions for each architecture under study. . . . .	189
TABLE 7.1.	FIPP <i>p</i> -cycle (DRS method) minimum capacity single span failure restorable designs: no <i>z</i> -case, 3000km reach feasible. . . . .	207
TABLE 7.2.	Total capacities of the architectures under study. . . . .	208
TABLE 7.3.	FIPP <i>p</i> -cycle (DRS method) equipment assignment results. . . . .	209
TABLE 7.4.	Number of working paths (out of 97) using same wavelength protection in FIPP <i>p</i> -cycle (DRS method) solutions. . . . .	210
TABLE 7.5.	Equipment costs of the architectures under study. . . . .	211
TABLE 7.6.	FIPP <i>p</i> -cycle minimum capacity single span failure restorable, non- <i>z</i> -case, reach limit feasible designs: FIPP DRS and FIPP MUPS IH solutions. . . . .	229
TABLE 7.7.	FIPP MUPS IH method solution configuration breakdown by types of paths protected and the minimum number of wavelengths required. . . . .	230
TABLE 7.8.	Equipment assignment results: Cost breakdown for FIPP MUPS IH and FIPP DRS solutions. . . . .	231
TABLE 7.9.	Number of working paths (out of 97) using same wavelength protec-	

tion in FIPP MUPS IH solutions. . . . . 232

TABLE 7.10. Equipment costs of the architectures under study, revised. . . . . 233

# List of Figures

---

FIGURE 2.1. The local, metropolitan and long-haul partitions of the transport network. . . . .	11
FIGURE 2.2. Examples of transport networks under study: (a) Germany network [10], (b) COST239 [11]. . . . .	13
FIGURE 2.3. Examples of graphs that are (a) connected, (b) two-connected and (c) bi-connected. . . . .	14
FIGURE 2.4. Possible service layer stacking in modern transport networks. . . . .	16
FIGURE 2.5. Add/drop multiplexer (ADM). . . . .	18
FIGURE 2.6. Optical cross connect (OXC). . . . .	19
FIGURE 2.7. Illustration of UPSR operation (a) before failure and (b) after failure. . . . .	32
FIGURE 2.8. Illustration of BLSR operation (a) before failure and (b) after failure. . . . .	34
FIGURE 2.9. Span restoration example showing three replacement paths used to restore three units of working capacity affected by failure. . . . .	36
FIGURE 2.10. Path restoration single failure example: (a) Two working paths prior to failure. (b) PR without stub release. (c) PR with stub release. . . . .	38
FIGURE 2.11. SBPP example. (a) Two working paths and unconnected spare capacity. (b,c) Two single span failure scenarios and the respective backup paths used. . . . .	39
FIGURE 2.12. DSP 3-way split example for a 7 unit demand bundle. . . . .	40
FIGURE 2.13. Example of a single PXT protecting 3 working paths. . . . .	42
FIGURE 2.14. Example of (a) a p-cycle and the action taken after (b) an on-cycle and (c) straddling span failure. . . . .	43
FIGURE 2.15. Dijkstra's algorithm Example. . . . .	66
FIGURE 3.1. a) On-cycle protection relationship and b) the restoration action taken after the working path fails. . . . .	72
FIGURE 3.2. a) Straddling protection relationship and b) restoration paths for a straddling route. . . . .	73
FIGURE 3.3. z-Case. FIPP p-cycle and working path are shown as a connected thick line and an arrowed line, respectively. . . . .	74
FIGURE 3.4. An example of a pure FIPP <i>p</i> -cycle configuration. . . . .	76

FIGURE 3.5.	Non-disjoint working paths sharing the same cycle for protection. a) Valid case: pre-failure state. b) Both working paths are restored. c) Invalid case: pre-failure state. d) Only one path restored because the protection paths provided by the cycle are non-disjoint. . . . .	78
FIGURE 3.6.	Path-protecting p-cycle example. a) Path set to be protected. b,c) Two cycles C1 and C2 assigned to protect the first two paths P1 and P2 respectively. c) P3 cannot be assigned a cycle for protection prior to failure.. . . .	79
FIGURE 4.1.	The FIPP IH algorithm.. . . .	93
FIGURE 4.2.	FIPP p-cycle DRS method. . . . .	96
FIGURE 5.1.	a) The data set generation flow model for the FIPP DRS JCP problem: (1) Eligible shortest routes set per node pair, (2) Disjoint route set candidates, (3) Eligible cycles for each DRS candidate. b) The data set and the model are combined and passed to the solver to yield a solution (outputs).. . . .	111
FIGURE 5.2.	GenerateDRSs pseudocode. . . . .	112
FIGURE 5.3.	Network cost vs. number of eligible working routes per demand with DRS history propagation.. . . .	121
FIGURE 5.4.	Lengths of eligible working route options relative to the shortest routes. (x, y) refers to the node pair that the route connects. . . . .	123
FIGURE 5.5.	Total network cost versus number of DRSs of various types. . . .	125
FIGURE 5.6.	Network cost vs. eligible route inclusion threshold with DRS history propagation. The best joint solution refers to the FIPP DRS JCP data point corresponding to 2 eligible routes per demand in Figure 5.3. . . . .	128
FIGURE 5.7.	DRS based SCP FIPP p-cycle solution (corresponding to the first data point of the FIPP DRS JCP solutions in Figure 5.3).. . . . .	131
FIGURE 5.8.	DRS based JCP FIPP p-cycle solution (corresponding to the second data point of the FIPP DRS JCP solutions in Figure 5.3). . . . .	133
FIGURE 6.1.	A four route DRS (arrowed lines) and the cycle originally generated to protect the DRS (solid thick connected line). The dashed arrowed line represents the over protected working path. (b) A possible outcome of actual protection assignment. . . . .	143
FIGURE 6.2.	Germany test network. (a) City view. (b) Span & Node identification view. . . . .	144
FIGURE 6.3.	Modified working routing between nodes 10 and 15. The dashed and solid lines represent the old and new route, respectively. . . . .	148

FIGURE 6.4.	Sample large configuration from the minimum capacity, non-z-case test set. . . . .	154
FIGURE 6.5.	(a) Histogram of total capacity values and (b) illustration of spans where the total capacity was 23 units for the solution obtained in Trial 3 using the modified model. . . . .	162
FIGURE 6.6.	Illustration of the 11 units of demand crossing the span between node 9 and 11. . . . .	164
FIGURE 6.7.	Transparent reach analysis: Minimum capacity solution (Trial 1, non-z-case.) . . . . .	168
FIGURE 6.8.	Transparent reach analysis: Minimum capacity, JCP, 20 wavelength solution. . . . .	170
FIGURE 6.9.	Transparent reach analysis: Minimum capacity, reach feasible solution. . . . .	172
FIGURE 6.10.	Nodal failures affecting on-span working routes. (a,c,e) represent the network state before failure and (b,d,f) represent the state of the network after failure. . . . .	176
FIGURE 6.11.	Node failures affecting straddling working routes. (a,c) represent the network state before failure and (b,d) represent the network state after failure. . . . .	177
FIGURE 6.12.	Nodal z-case: a straddling working route that is not node disjoint from the protection paths provided by the cycle. (a) is the network state before failure and (b,c) are examples of possible network states after failure. . . . .	178
FIGURE 6.13.	Mutual capacity scheduling conflict occurring as a result of node non-disjoint working routes sharing the same cycle for protection. (a) is the network state before failure and (b,c) are examples of possible network states after failure. . . . .	179
FIGURE 7.1.	Nodal structure of a 2 input fiber port WB based transparent ROADM. Unidirectional schematic shown. Adapted from [77]. . . . .	197
FIGURE 7.2.	Nodal structure of a WSS based transparent OXC. Unidirectional schematic shown. Adapted from [77]. . . . .	198
FIGURE 7.3.	Span equipment implementation. Unidirectional schematic shown. . . . .	199
FIGURE 7.4.	Generalized node model. Bidirectional schematic shown. . . . .	201
FIGURE 7.5.	End-node implementation of a working path under different wavelength protection. (a) Pre-failure state. (b) Failure state. . . . .	202

FIGURE 7.6. End-node implementation of a working path under same wavelength protection. (a) Pre-failure state. (b) Failure state. . . . .	203
FIGURE 7.7. Implementation of an end-node of a straddling working route where one of the working paths is protected using same wavelength protection. (a) Pre-failure state. (b) Failure state. . . . .	204
FIGURE 7.8. The FIPP_MUPS_IH algorithm. . . . .	215
FIGURE 7.9. GenerateWorkingRoutesMaxStraddlers algorithm. Used to generate working routes for every demand relation such that the total number of potential straddling relationships is maximized. . . . .	216

# List of Abbreviations

---

ADM	Add/Drop Multiplexer
AMPL	A Mathematical Programming Language
APS	Automatic Protection Switching
ATM	Asynchronous Transfer Mode
B&B	branch-and-bound
B&S	Broadcast and Select
BLSR	Bidirectional Line-Switched Rings
BER	Bit Error Rate
CG	Column Generation
CIDA	Capacitated Iterative Design Algorithm
CO	Central Office
CWDM	Coarse Wavelength Division Multiplexing
DCF	Dispersion Compensating Fiber
DFS	Difficult Share First
DGE	Dynamic Gain Equalizers
DP	Diverse Protection
DRS	Disjoint Route Set
DS	Digital Signal
DSL	Digital Subscriber Line
DSP	Demand-wise Shared Protection
DW	Digital Wrapper
DWDM	Dense Wavelength Division Multiplexing
E	European
EXC	Electrical Cross-Connect
FCC	Federal Communications Commission
FCFS	First Come First Serve
FIPP	Failure Independent Path Protection
FP	Full Protection
GA	Genetic Algorithm
GFP	Generalized Framing Protocol
GMPLS	Generalized Multi-Protocol Label Switching

HAVANA	High-Availability Network Architectures
IA	Inline Amplifier
IH	Iterative Heuristic
IJD	Iterative Joint Design
ILP	Integer Linear Programing/Integer Linear Program
IP	Internet Protocol
JCP	Joint Capacity Placement
LAN	Local Area Network
LHN	Long Haul Network
LP	Linear Programing/Linear Program
MAN	Metropolitan Area Network
MIP	Mixed Integer Programing/Mixed Integer Program
MTD	Maximum Transmission Distance
MUPS	Maximum Unit Path Straddler
Mx/Dx	Multiplexer/Demultiplexer
NEPC	Node Encircling $p$ -Cycle
OA	Optical Amplifier
OADM	Optical Add/Drop Multiplexer
OC	Optical Carrier
o-e-o	Optical-Electronic-Optical
o-o-o	Optical-Optical-Optical
OPPR	Optical Path Protection Rings
OR	Operations Research
OSPR	Optical Shared Protection Rings
OXC	Optical Cross-Connect
PDH	Plesiochronous Digital Hierarchy
POP	Point of Presence
PPS	Path Protected Set
PR	Path Restoration
PVWP	Partially Virtual Wavelength Path
PXT	Pre-Cross-Connected Trail
QoP	Quality of Protection
RHS	Right Hand Side

ROADM	Reconfigurable Optical Add/Drop Multiplexer
RWA	Routing and Wavelength Assignment
R1	Single-Failure Restorable
R2	Dual-Failure Restorable
SBPP	Shared Backup Path Protection
SCP	Spare Capacity Placement
SDH	Synchronous Digital Hierarchy
SONET	Synchronous Optical Network
SPE	Synchronous Payload Envelope
SR	Span Restoration
STM	Synchronous Transport Modules
STS	Synchronous Transport Signal
TDA	Transparent Node Amplifier
TSP	Traveling Salesman Problem
UPSR	Unidirectional Path Switched Rings
VT	Virtual Tributary
VWP	Virtual Wavelength Path
WAN	Wide Area Network
WB	Wavelength Blocking
WDM	Wavelength Division Multiplexing
WP	Wavelength Path
WSS	Wavelength Selective Switch
XPDR	Transponder

# Chapter 1: Introduction

---

## 1.1 Introduction

The Internet is undoubtedly one of humanity's most important and complex systems and this importance will only grow as more Internet services and applications are developed. Connecting over a billion users and prevalent in nearly every sphere of human activity, it is built and fundamentally dependant on the underlying backbone transport network. In addition to Internet traffic, this network also services cellular phones, banking machines, leased lines, etc. This network handles a tremendous amount of data. Technologies such as dense wavelength division multiplexing (DWDM) allow for transmission of multiple terabits per second of traffic in a single fiber. Like any other man-made system, the transport network is susceptible to failure and as more is demanded of it, and as more mission critical services are deployed, the more crucial it becomes to protect this network and the traffic on it.

The physical medium that allows for the backbone network to handle the large amount of data that it does is the fiber optic cable. This medium is very fragile and despite the vast amount of resources spent on maintenance and preventive measures, cable cuts are the most common reason for transport network outages. The reasons for these failures range from natural causes to human error and sometimes even sabotage. Cables placed underground are vulnerable to dig ups and rodent damage, as well as landslides and floods. Aerial cables are subject to vehicle damage by snagging or by accidents involving telephone poles. They are also vulnerable to fires, tree falls, contact with power lines and ice. Undersea cables are vulnerable to being tangled in fishing nets, anchors dragging across the ocean floor, currents and tides rubbing the cable on sharp rocks on the ocean floor as well as seismic activity. According to a study done by the Federal Communications Commission (FCC) in 2002, it is estimated that for a 1000 miles of cable, a long haul network experiences 3 cuts annually, whereas metropolitan networks experience as many as 13 [1]. This means that a typi-

cal longhaul network with 30,000 miles of fiber can expect an interruption from a severed cable every 4 days or so and a network containing 100,000 miles of cable will experience an interruption more than once a day. Severed optical cable is not the only cause of transport network outages. Network equipment, support and power systems are also susceptible to failure. These systems, however are housed in protected and maintained facilities with redundant network equipment, backup power supplies and are staffed by technicians who are able to address problems quickly when they arise.

To the end-user the majority of these failures are virtually invisible. Enough spare resources are provisioned so that affected traffic is quickly re-routed through the network ensuring that connectivity is maintained while the repairs are underway. Despite best efforts, major outages still occur and are quickly picked up by the media, often as front page news. The most notorious event in recent history happened in February 2008, when four undersea cables failed in the Mediterranean region over a course of a week due to cables being severed by anchor dragging and disruption due to power failure. The events resulted in massive outages in Egypt, the Middle East and India [2][3][4]. Another recent example happened in the same region in December 2008, when three cables were almost instantaneously severed by seismic activity [5]. In November 2003, the TAT-14 ring system linking Europe and the United States suffered a major disruption when a fault occurred in an undersea portion of the cable between France and the Netherlands, causing disruption of service in the UK. This system was vulnerable at this time because part of the system near the US coast already suffered a fault earlier in the month and was not yet repaired [6]. In the United States in recent years, heavy rains, leading to a washout between Palm Springs, California and Phoenix, Arizona, caused two cables to be cut resulting in service disruption for thousands of customers [7]. Another example happened in southern Colorado where the heat from a controlled burn melted underground fiber optic cable, causing a 9 hour outage in the region [8].

The extra capacity required to guarantee that the network can be reliably restored after failure without the end-user noticing is not an unlimited resource. Laying tens of thousands of kilometers of cable is capital intensive as is the equipment necessary to support 10Gbits/s (OC-192) line rates commonly deployed in modern transport networks. Because of this, it is in the interest of the network operators to deploy network protection that is efficient and cost effective. To date, several different network protection architectures have been introduced, each providing unique advantages such as speed of recovery, capacity/cost efficiency, operational simplicity, ease of implementation and others. One particularly new protection architecture, and one that is the focus of this thesis, is called failure independent path-protecting (FIPP)  $p$ -cycles, introduced below.

## 1.2 Failure Independent Path-Protecting $p$ -Cycles

Recently, a new architecture was introduced called failure independent path-protecting (FIPP)  $p$ -cycles ([38], [39], [40]). It extends the concept of span-protecting  $p$ -cycles to end-to-end path protection. Being an architecture that utilizes cyclical protection structures, similarly to ring systems and  $p$ -cycles, it allows for pre-connection of protection paths. This means that these paths may be established and in a fully ready and tested state before the onset of failure. This is a desirable property in transport networks and is related to the speed at which the architecture can recover traffic affected by network disruptions. The property of pre-connection is especially important in optical networks where it is unrealistic to expect the quality of the protection path to be high if it has to be assembled and cross-connected in real time, immediately after failure. The reason for this is because, in optical transparent networks, the signal cannot be regenerated as it travels from its origin to its destination. Instead, the signal must be cross-connected optically end-to-end and the integrity of the entire path (as opposed to inter-repeater path sections) must be designed for. This architecture allows for capacity efficiencies that are in the range achieved by shared backup path protec-

tion (SBPP); a state of the art path-oriented network protection architecture that does not utilize pre-connection. Lastly FIPP  $p$ -cycles allow for simple, end-node activated, failure independent restoration actions making it a unique architecture on which there has been relatively little research done, compared to span-protecting  $p$ -cycles and SBPP. Because of the properties mentioned above FIPP  $p$ -cycles is a strong contender for implementation on optical transport networks.

### 1.3 Thesis Outline and Highlights

This thesis advances the state of the art of the FIPP  $p$ -cycle concept, contributing significantly to the FIPP related body of literature. The body chapters of this thesis are organized into three main sections. Chapter 2 and Chapter 3 introduce the background concepts relevant to transport networking and also introduce the FIPP  $p$ -cycle concept in detail. Chapter 4 and Chapter 5 focus on new methods developed for generating FIPP  $p$ -cycle solutions. Chapter 6 and Chapter 7 outline the FIPP  $p$ -cycle studies done as part of the collaboration project between TRILabs (Edmonton, Alberta, Canada) and Nokia Siemens Networks (Munich, Germany). This collaboration project, referred to as HAVANA (which stands for High AVAILability Network Architectures), was a commissioned study that focused on four main pre-connected architectures: FIPP  $p$ -cycles, pre-cross-connected trails (PXTs), demand-wise shared protection (DSP) and span-protecting  $p$ -cycles. This project spanned 3 years (November 2005 - November 2008) and the main focus of the project was to perform investigations on the pre-connected architectures in the context of transparent optical networking under specific network equipment constraints.

Chapter 2 introduces the relevant background topics setting the stage for the rest of the thesis. The field of transport networking, including the equipment and the technologies used, is introduced in detail. The concept of network failure and survivability is examined in closer detail and the chapter includes a thorough summary of the state of the art transport network survivability architectures. This chapter presents

a review of graph theory, optimization and related mathematical algorithms relevant to survivable transport network design.

Chapter 3 introduces the FIPP  $p$ -cycle concept in detail, including a thorough discussion on the types of different working route/cycle relationships and how a set of working routes sharing a cycle for protection relate to one another and to the cycle. It also contains a thorough literature review of the FIPP  $p$ -cycle concept.

Chapter 4 introduces two new methods for FIPP  $p$ -cycle design: namely FIPP column generation (CG) and FIPP iterative heuristic (IH). The prior is a new method that utilizes the column generation technique for solving linear program (LP) problems. Because of how the individual cycle/route set structures are generated, this method allows for non-joint working paths to be protected by the same cycle. This allows FIPP CG to generate solutions that have the highest capacity efficiency relative to all other known FIPP methods. FIPP IH is a fast method that iteratively generates custom working route sets to be protected by the cycles found in the network such that the specified credit score is maximized. Additionally, Chapter 4 reviews the FIPP disjoint route set (DRS) method, which is used extensively in this thesis. The solutions generated by the three methods are presented and contrasted against one another with FIPP CG yielding the most capacity efficient solutions and FIPP IH yielding solutions that are comparable to the DRS method.

Chapter 5 proposes an extension to the FIPP DRS method that allows for multiple distinct shortest working routes per demand relation to be considered where as the original method only considered the shortest working route. The new method, called FIPP DRS joint capacity placement (JCP) is demonstrated to have considerable savings relative to the original spare capacity placement (SCP) method (in the order of 13%). JCP solutions are also shown to be within 4% of optimal span-protecting  $p$ -cycle solutions. Additionally, a threshold effect is observed, where the JCP method is shown to yield the most significant gains when using only two distinct shortest route

options per demand relation. Additional shortest route options are shown not to provide significant additional improvements.

Chapter 6 contains a series of FIPP  $p$ -cycle studies performed in the first two years of the HAVANA project. This chapter starts with an initial exercise in which minimum capacity single span failure restorable designs are generated for every architecture under study. The goal of this exercise is twofold: The first is to enable an initial comparison between the architectures under study and the second is to establish a base set of solutions that can be used in further investigations. The first study involving these solutions is an investigation into how the different initial single failure restorable designs perform when affected by dual span failure scenarios. This study yields two potential dual-failure restorability strategies available to FIPP  $p$ -cycles. The next study deals with minimum wavelength assignment under specific equipment constraints. These constraints set the maximum number of wavelengths available at 40, far above the actual minimum number of wavelengths required by the minimum capacity solutions FIPP  $p$ -cycles, which is 23. Under the equipment assumptions made, lower cost equipment is available for designs requiring a total of 20 wavelengths or less and the latter part of the wavelength assignment section is dedicated to meeting this goal. It is reached by generating new minimum capacity solutions using the FIPP DRS JCP method, which was introduced in Chapter 5.

The third study in Chapter 6 introduces further network equipment constraints that set a 2000 km limit on the transmission distance of any working or protection path that must be adhered to before the design can be implemented. This study starts by first evaluating how well the initial minimum capacity and maximum 20 wavelength solutions adhere to the transmission reach limit. It turns out that the reach limit is exceeded by 45.4% of the protection paths in the initial FIPP solution and by 57.7% of the protection paths in the 20 wavelength solution. This study continues by generating a new minimum capacity design where all the paths are as short or shorter than 2000 km; a goal that is reached at the price of a 22% increase in spare

capacity relative to the initial minimum capacity solutions. Nodal failure is the subject of the final study in this chapter. A first step in this study involves evaluating how the initial minimum capacity solutions perform in light of node failures and two strategies are proposed for coping with them. This is followed by an introduction of a revised FIPP DRS method that is capable of generating 100% node failure FIPP  $p$ -cycle designs. It is found that the solutions generated by this method provide 100% node failure restorability at the cost of a 13.2% increase in spare capacity relative to the initial minimum capacity single span failure restorable designs.

Chapter 7 contains the set of studies performed in the third and final year of the HAVANA project. The main goal of this chapter is to investigate how the architectures under study may be implemented in a real optical transport network and how the different architectures compare with one another in terms of capital expenditure needed. The chapter starts out by introducing the NOBEL cost model, which outlines the relative equipment costs. This cost model is based on the NOBEL project in Europe that was aimed at determining the relative costs of transport network equipment. Specifically, this model specifies the cost of equipment used in opaque and transparent optical networking, making it very useful for the HAVANA project. The NOBEL cost model imposes a new set of maximum transmission distances, the upper limit of which is 3000 km. First new minimum capacity solutions are generated to adhere to this reach limit, and the NOBEL model is applied to these solutions. It is found that FIPP  $p$ -cycle solution implementations are very costly, relative to the other architectures. A closer look reveals that minimum capacity solution implementations do not utilize same wavelength protection very well, the use of which is highly cost effective under the NOBEL cost model. In light of this, a new method that maximizes the number of same wavelength protection relationships is introduced. This method, called the FIPP maximum unit path straddlers (MUPS) method, is then used to generate FIPP  $p$ -cycle solutions that show a significant reduction in cost, relative to the 3000 km limited initial minimum capacity solutions (in the order of 23%).

Chapter 8 concludes the thesis, summarizing its results and contributions and providing closing remarks.

# Chapter 2: Background

---

## 2.1 Introduction

This chapter provides a general overview of transport networking and the necessary background to supplement the ideas presented in this thesis. Basic network-modeling concepts are introduced first. This is followed by a closer look at the network equipment and then by an introduction of network technologies relevant to this thesis. Next, the concept of survivability and related terminology is introduced, which is followed by a comprehensive introduction to the different types of protection and recovery schemes that have been invented for transport networking. Lastly this is followed by a section on relevant mathematical concepts including linear programming and algorithms used in this thesis. Closer examinations of the topics covered in this chapter can be found in Chapter 1 to Chapter 3 of [9].

## 2.2 The Transport Network

A transport network, also known as a backbone network, is primarily responsible for efficient, fast and reliable point-to-point data transmission. Residing just above the physical equipment layer, it is a resource for services such as the Internet, leased lines, cellular phones, bank machines, and 911 calls. The data from these services is multiplexed together and routed over an infrastructure of logical point-to-point multi-channel transmission systems to common destinations using high data rate signals. A real world analogy of the role that a transport network plays can be found at the post office. Sent letters and parcels are not delivered individually from the sender's doorstep to the recipient's mailbox. Instead, all the letters from the sender's neighborhood are collected into mailboxes, which are then taken to the local post office depot and then to the processing plant where the letters and parcels are sorted according to common destination. They are delivered to processing plants close to their destinations in bulk by airplane, truck or train much in the same way that a

transport network delivers aggregations of traffic to their respective common destinations. Once at the destination processing plants, the letters are then sorted and sent to local post office depots. They are then sorted again and placed in neighborhood letter boxes from where they are delivered to their final destinations by postal workers. Ultimately, a transport network is not concerned with details such as the exact origins or destinations of the services it carries. Its function is to efficiently fill the high data rate containers and to deliver these reliably to a common destination hub where the services using the transport network are demultiplexed and sent to their final destinations.

## **2.3 Transport Network Hierarchy**

A transport network can be partitioned conceptually into a three level hierarchy based on geographic, political and/or administrative boundaries as shown in Figure 2.1. The three levels are the local area network (LAN), the metropolitan area network (MAN) and the long-haul network (LHN). The LAN is typically characterized by a wide variety of access mechanisms and protocols. A brief list of services that access the transport network through the LAN include cell phones, corporate wide area networks (WANs) as well as residential customers using Digital Subscriber Line (DSL), cable or dial-up. The data from these services is aggregated at the local switching offices/hubs and is routed onto the larger MAN. The MAN is a network composed of several regional central offices and hubs connected together. The span distances in such networks are generally less than 50km and the cost of such networks is dominated by the cost of nodal equipment. It is possible for several MANs to exist in the same city and they interface with one another at access points called points of presence (POPs). Data originating at one MAN and destined for another is exchanged at these points. Data destined for other cities is aggregated and routed onto a LHN which connect several metropolitan networks on a national/international level. Because of the long distances between nodes in a network (reaching 1000s of kilome-

ters), the main cost of the network comes from cable costs and installation as well as the cost of repeaters, regenerators and amplifiers. LHNs also interface with one another at points of presence (POPs), and it is not uncommon for data to traverse several networks owned by separate entities before arriving at its destination. In line with the post office example from before, the LAN analogous to the post offices and the postal workers. The MAN is analogous to the set of depots in a single region and the LHN is analogous to the set of processing plants.

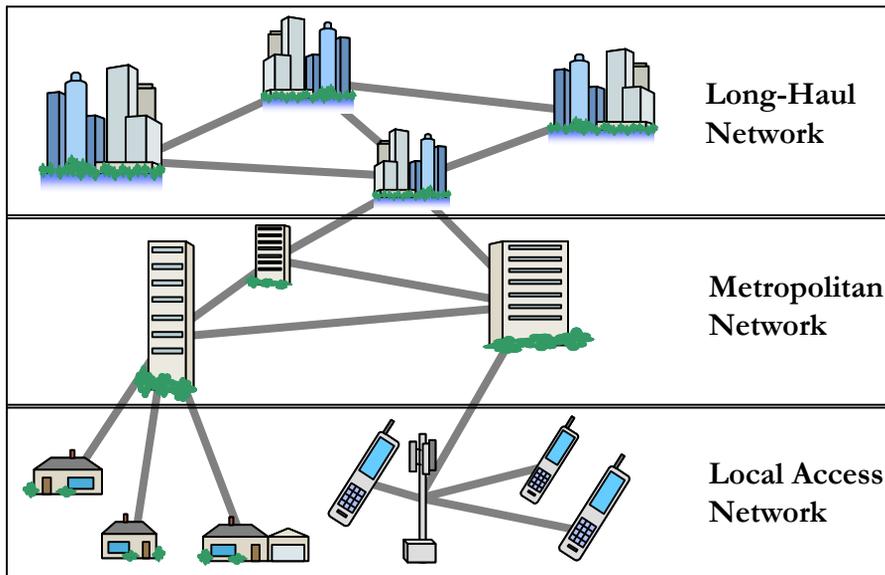


FIGURE 2.1. The local, metropolitan and long-haul partitions of the transport network.

## 2.4 The Network Model

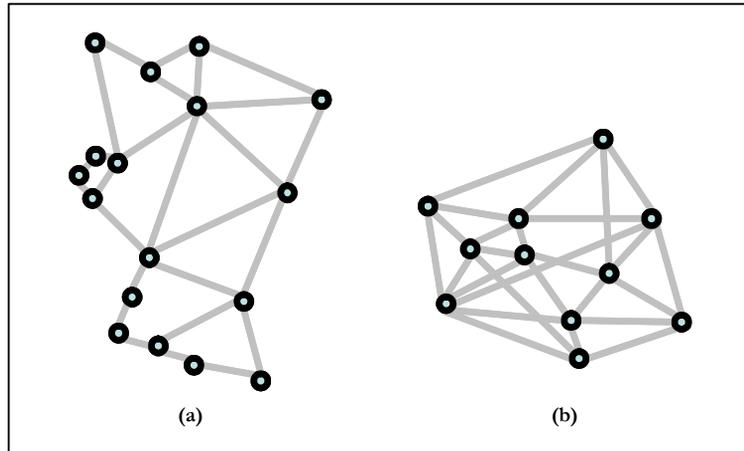
A transport network, at the top-most level, can be modeled as a collection of nodes and spans. A node in a network graph represents access points to the network. Nodal equipment such as optical add/drop multiplexers (OADMs) and optical cross-connects (OXC) are housed at these points, which act as origin and destination points for transport network traffic. A span represents a direct connection between

equipment at two different nodes. In transport networking a span generally represents lengths of fiber optic cable connecting nodal equipment. A terms span and node are the transport networking equivalent to an edge and vertex from graph theory, respectively.

A link, or channel, describes a basic unit of capacity between nodes sharing a span. What this actually corresponds to depends on the context and can be an optical carrier (OC-n) signal in the synchronous optical networking (SONET) context, a wavelength on a span in a wavelength division multiplexing (WDM) context or a light path (meaning a single wavelength used for a path end-to-end), for example. A single span can contain one or more links or channels. In the context of this work, an individual link is considered to be bidirectional meaning that an allocated unit of capacity allows for transmission from node A to node B and vice versa, given that the node A and B are neighbour nodes connected by a span containing the link. A route is defined as a concatenation of spans that geographically connect two nodes. A path describes a logically cross-connected sequence of individual links going over those spans. Since the links making up the path are bidirectional, the end-to-end path is considered to be bidirectional as well.

The collection of spans and nodes, as well as the ways in which they relate to one another make up what is referred to as a network topology. The ways in which nodes and spans connect to one other can be described using an adjacency matrix. In a transport network, a span can connect to exactly two different nodes. Two nodes connected by a span are referred to as adjacent nodes and the span connecting two adjacent nodes is adjacent to those nodes. A single node may have one or more adjacent spans. The number of spans adjacent to a node is referred to as the nodal degree. The sum of all the nodal degrees divided by the number of nodes in the network results in what is called the average nodal degree. This metric is useful for classifying the general connectivity of a network. Typical transport networks have an average nodal degree between 2.5 and 5. For example, the two main networks used in this the-

sis, the Germany [10] and COST239 European networks [11] (used as a regional representative transport network planning models) are illustrated in Figure 2.2 and have nodal degrees of 3.1 and 4.7 respectively.



**FIGURE 2.2.** Examples of transport networks under study: (a) Germany network [10], (b) COST239 [11].

The network topologies used to model transport networks take on the shape of simple graphs as defined in the field of graph theory. The parallels to node and span in the subject of graph theory are vertex and edge, respectively and these terms are used interchangeably. A simple graph, by definition, does not contain any self-loops or parallel edges. A self-loop is an edge that starts and ends at the same vertex. Parallel edges refer to two or more non-self-looping edges that share vertices. Network topologies used in survivable transport networking have the graph connectivity property of bi-connection. Such network graphs, illustrated by Figure 2.3c, will not have any stub nodes nor articulation points. A stub node is a node that has a nodal degree of 1. Stub nodes are undesirable since a failure affecting the only span connecting the stub node to the network will isolate it completely, rendering service recovery impossible. An articulation point (also called a bridge/cut node) is a node that, if removed, dissects the network completely into two disconnected parts. A graph that contains at least one stub node is called a connected graph as illustrated by Figure 2.3a. A network that

does not have any stub nodes but does have articulation points is called two-connected, illustrated by Figure 2.3b. A two-connected graph is undesirable because a node failure at the articulation point makes it impossible to recover any of the affected connections. Another way of looking at bi-connectivity is that a bi-connected network graph cannot be disconnected into two parts by removing any one node or span. In practice, virtually all survivable fiber based transmission systems are bi-connected.

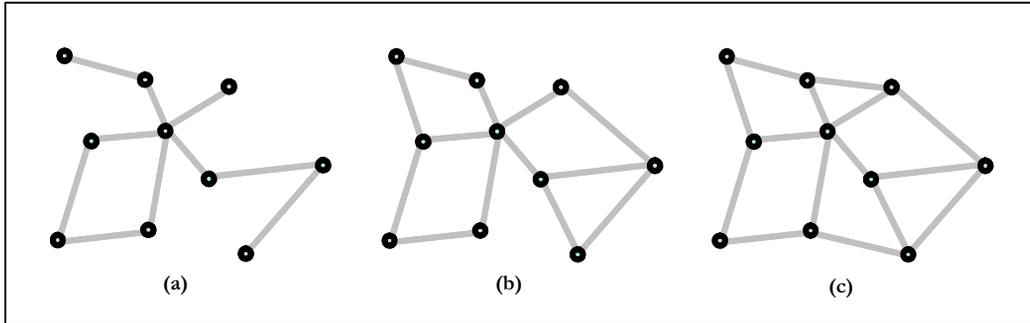


FIGURE 2.3. Examples of graphs that are (a) connected, (b) two-connected and (c) bi-connected.

## 2.5 Virtual Logical Networks

The physical backbone network exists as a combination transmission devices such as OXCs, OADMs and support equipment such as amplifiers, regenerators and equalizers connected together by fiber optic cable. This physical fiber transmission system can be cross-connected to create a number of logical transport configurations. Cross-connection refers to the interconnection of point-to-point channels in the physical graph. The patterns created by cross-connecting point-to-point channels are referred to as virtual topologies. On any given transport network it is possible to have as many unique virtual topologies as there are ways to cross-connect the individual transmission channels.

The underlying physical network is rarely seen directly by the services using it. Instead, the services see virtual logical abstractions/topologies which are perceived as dedicated networks to the services using them. These logical networks, presented as logical arrangements of carrier signals over fiber optic links, remove the unnecessary details from the view of the users. From the perspective of the services, the transport network is transparent. Meaning that the services may use any rate or type of payload and not encounter any synchronization problems so long as the transport network has an established connection between points A and B. Being able to logically reconfigure the network allows network operators to adapt the network to the services using it on a relatively short time scale. In contrast, physically altering a network is a time and capital intensive task. Because of this the physical network is altered and upgraded based on future demand forecasts; when it becomes apparent that the physical network will be unable to support the forecast future logical configurations.

The services utilizing the transport network are not restricted to being serviced by the underlying physical layer directly. The services themselves can be seen as layers in the underlying transport network and can stack on top of one another with the lower service topologies in the stack acting as transport networks for the higher ones. Each layer has a set of important standardized functions that can, for example, allow ethernet traffic to be applied to a laser for optical transmission over long distances using the underlying transport network; a task that Ethernet traffic is not by itself suitable for. Figure 2.4 shows a small number of possible service layer stacking arrangements. For example, it is possible for packet data from the internet protocol (IP) [12] layer to be serviced by the asynchronous transfer mode (ATM) [13] layer, which is serviced by the SONET [14] layer, which, in turn is serviced by the Dense Wavelength Division Multiplexing (DWDM) [15] layer, which is then serviced by the underlying physical layer. It is also possible to have the IP layer serviced by the Generalized Multi-Protocol Label Switching (GMPLS) [16] layer which is then serviced by the DWDM and the physical layer. The amount of stacking that occurs in a net-

work is typically dependant on the needs of the applications that require service as well as the hardware implemented in the network. Furthermore, as network technology evolves, layer stacking is useful for preserving legacy standards that may be cost prohibitive to upgrade. The details relating to the service layers shown in Figure 2.4 are out of the scope of this thesis but can be found in [9].

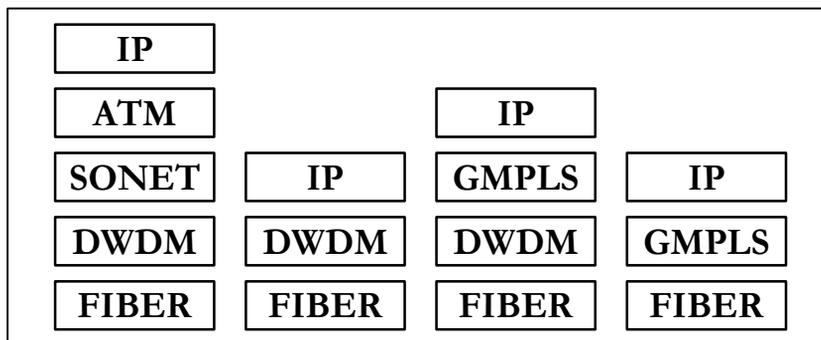


FIGURE 2.4. Possible service layer stacking in modern transport networks.

## 2.6 Demand

Demand refers to the aggregated capacity and routing requirements imposed on the network by the upper level service layer traffic flows. There is a distinction between traffic and demand: traffic refers to measures of data flow intensity (tied to a specific type of communication such as voice, video, etc.) where as demand refers to quantum of transmission capacity required to support the aggregated traffic flows. Demand is the measure of how many containers are required to be available between a pair of nodes so that all the required data is delivered to its destination and in the context of this thesis, is assumed to be bidirectional. Demand may take on standard transmission units such as light paths, OC-n or digital signal (DS-n) signals, for example.

In this thesis, one unit of demand corresponds to a request for a unit lightpath between the demand's end-nodes. The network is assumed to be using DWDM tech-

nology meaning that the links on a span correspond to wavelengths to be used for transmission and each unit lightpath reserves a single wavelength on every span it crosses. Multiple demands between the same node-pairs are referred to as a demand bundle. Demand quantities between nodes, are organized into what is called a demand matrix, which gives the total demand requested of a given network. This matrix is then used to perform working capacity routing, which influences spare capacity placement, as will be seen later on.

## **2.7 Network Equipment**

Up to this point, the transport network has been presented in terms of nodes and spans, which are abstractions for a variety of components that are key in the operation of the network. Recall that the node represents an central office (CO) building, complete with the electrical and line terminating systems contained within it whereas the span represents the fiber and the span equipment connecting the nodal devices. While detailed examinations of many of the transport network devices are out of the scope of this thesis, this section introduces some key components with the intent of giving the reader a brief introduction to what goes on ‘under the hood’ in the transport network. Additional information may be found in [9] and [17].

### **2.7.1 ADM/OADM**

An add/drop multiplexer (ADM) is a device that is used at the nodes of the network. It has two optical interfaces that can accept and transmit line rate signals as shown using a simplified illustration in Figure 2.5.

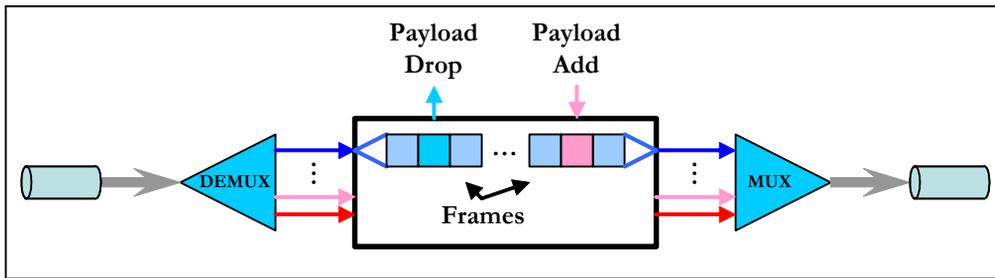


FIGURE 2.5. Add/drop multiplexer (ADM).

Internally, these devices are capable of demultiplexing individual incoming lightwave channels and electronically detecting payloads such as SONET/SDH frames carried on these wavelengths. Demultiplexing refers to the process of separating the individual wavelengths from the signal incoming from the fiber into the device. Conversely, multiplexing refers to the process of taking several different wavelengths and converting them into a single signal so that it can be sent out on an outgoing fiber. The payloads arriving at the ADM can then be either dropped at the node or electronically passed through to an appropriate outbound port. New payloads can also be added electronically. The newly added and pass-through payloads are modulated onto appropriate optical carriers which are then multiplexed together and transmitted.

The optical version of an ADM is called an optical add/drop multiplexer (OADM). Where the signals in an ADM undergo optical-electronic-optical (o-e-o) operations described, signals in an OADM remain in the optical (o-o-o) domain. The payloads added and dropped by an OADM are the individual wavelengths and these payloads may also be optically passed-through the device. A more specialized version of an OADM is a reconfigurable optical add/drop multiplexer (ROADM) which is a device that can also be remotely reconfigured where as regular OADMs do not have this functionality.

## 2.7.2 OXC

An optical cross-connect (OXC) is a line-terminating device designed to cross-connect incoming and outgoing channels. All the signals going into an OXC are handled simultaneously, and for this reason are referred to as non-blocking space switches. The general OXC architecture is illustrated in Figure 2.6.

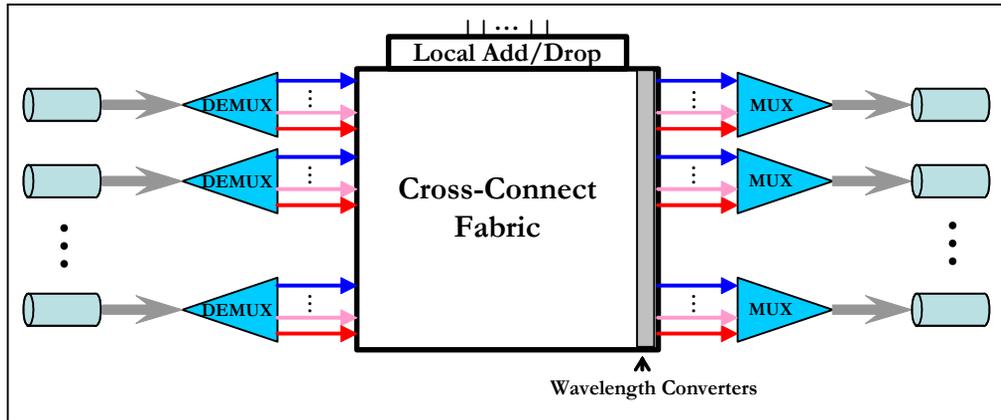


FIGURE 2.6. Optical cross connect (OXC).

Main components of an OXC include input and output fiber ports, multiplexer/demultiplexer units to separate and combine carrier wavelengths and at the heart of the device, a cross-connecting switch fabric also called the switch core. Modern commercial OXCs are realized using electronic switch cores interfaced by transponders that convert incoming optical signals to the electronic domain and convert outgoing electronic signals to the optical domain. Implementation of this type architecture is capital intensive as well as costly in terms of space and power, especially considering that OXCs are designed to interface with hundreds of optical fibers. The conversion to the electrical domain is a limiting factor in the amount of traffic an OXC can handle and while the conversion to the electrical domain does provide some benefits in terms of regeneration and network operation, it may not be desirable at every node. There is ongoing research into developing commercial devices that have the capability to optically cross-connect signals at the wavelength level. These devices

can either be completely optical or partially optical, giving rise to transparent/translucent networks as discussed below.

Networks built using OXCs that electronically cross-connect signals and ADMs are referred to as opaque or virtual wavelength path (VWP) networks. These networks, while costly, allow for regeneration and wavelength conversion as well as for failure isolation functions and network administration. Networks built using all-optical OXCs and OADMs are called transparent networks or wavelength path (WP) networks. These networks optically connect end-to-end channels between nodes. This has several benefits including lower cost and power relative to opaque networks as well as independence of the network to the payloads applied to the lightpath channel connecting the end-nodes. This property of payload transparency is highly desirable because it reduces the number of encapsulation layers needed in a transport network by allowing the services to communicate using their native protocols. Protocols such as digital wrapper (DW) and generalized framing protocol (GFP) have been developed to support payload transparency in opaque networks.

Since at this time, commercial technology for all-optical regeneration and wavelength conversion does not exist, transparent transport networks lack this functionality. Not allowing for regeneration at intermediate nodes may limit the transmission length between end-nodes and without wavelength conversion a network is susceptible to the routing and wavelength assignment (RWA) problem and may encounter related wavelength blocking issues. Briefly, the RWA problem refers to the difficulty associated with finding lowest cost working routing between end-nodes under the constraint that the optical paths must use a single wavelength, end-to-end. Wavelength blocking refers to the case where enough capacity exists between end-nodes, but the lightpath request cannot be satisfied because all the potential paths violate the single wavelength path constraint. The compromise between opaque and transparent works is a translucent network, otherwise known as a partially virtual wavelength path (PVWP) network. These networks are realized using a combination

of all-optical and strategically placed electronic core OXCs or by using PVWP OXCs at every node; optical devices that allow for electronic regeneration and wavelength conversion for a subset of wavelength paths.

### **2.7.3 Optical Amplifiers and Regenerators**

Nodal equipment such as OXCs and OADMs are rarely connected just by fiber cable for distances longer than 60km. At this distance and higher, attenuation and other effects degrade the signal past levels acceptable for reliable transmission. The optical amplifier (OA) allows for transmission beyond this limit by boosting the complete band of lightwave channels optically, without individually processing each individual carrier. The OA is an analog amplifier and, similar to electronic amplifiers, boosts the useful signal to overcome attenuation and insertion losses, but also boosts the noise and propagates errors in the signal. Furthermore, the OAs themselves are not noiseless nor perfectly linear. At distances of about 550km the above factors accumulate to the point where optical amplification is not enough to significantly increase the transmission distance.

Regenerators are used to increase the transmission distance beyond what is possible using optical amplifiers only. These devices demodulate the individual signal from their optical carrier and convert it to the electrical domain. The signals are inspected to determine their state and then new 0/1 symbols are generated and modulated back onto the optical carrier wavelength. This process is referred to as 3R regeneration: re-timing, regeneration and retransmission. Compared to optical amplification, regeneration is very costly because it involves o-e-o conversion and high-speed electrical processing of each individual channel. Regeneration can also be performed at intermediate nodes along the route of a path by OXCs and OADMs. Stand alone regenerators are only used in long haul transmission routes and are significant contributors to cost of these facilities.

## 2.8 Transmission Technology

A typical span in a network graph represents a collection of point-to-point transmissions systems between two nodes. This is typically a set of cables routed together, containing optical fibers used to connect nodal equipment at adjacent nodes together. This section introduces two of the most significant transmission technologies used in modern transport networks: SONET and DWDM. Together, these technologies allow for multiple Tbits/s of data to be carried on a single fiber.

### 2.8.1 SONET/SDH

The synchronous optical network (SONET) standard ([14],[18],[19]) is the most widely used optical-based transport networking standard in the world today. It emerged in 1987 in response to growing disparity between propriety equipment and transmission standards used by competing telecommunications companies. This new standard ensured the compatibility of the networks and equipment of competing companies. It defined the functional models for all elements of the transport network such as standard data rates, framing, acceptable bit error rates (BER), line cards as well as digital/optical cross-connects and multiplexers. It allowed for the body of knowledge on transport networking accumulated up to that point to be implemented. It allowed byte oriented synchronization that allowed for direct access to digital signal-0 (DS-0) signals without multiple demultiplexing. The overheads introduced into the network transport signals under SONET allowed for higher degree of robustness as well as made the network more amenable to survivability.

Briefly, the DS-0 is a tributary level data frame corresponding to a single digital capacity channel that originated with the first digital carrier system: the T-1 carrier system. These DS-0 signals were interleaved in sets of 24, 8 bits at a time to make the DS-1 signal, which was used on each physical T-1 carrier system line. Under the DS-1 protocol, transmission rates of 1.544 Mbits/s were on each carrier line were achieved.

As newer technology made faster digital transmission rates possible, further digital signal multiplexing was used to get more out of those transmission rates. A DS-2 signal was created by multiplexing 4 DS-1 signals together and 7 DS-2 signals can then be multiplexed to make a DS-3 signal and so on. Under DS-2 and DS-3 transmission rates of 6.312 Mbits/s and 44.736 Mbits/s were achieved, respectively. This multiplexing hierarchy is referred to plesiochronous digital hierarchy (PDH) and was widely used on twisted pair, microwave as well as first generation optical fiber systems. Under PDH, in order to access or drop individual tributary DS-0 signals from a DS-3, for example, the DS-3 must first be demultiplexed into DS-2s, then into DS-1s and only then can the DS-0 be accessed. The reverse is true if a DS-0 needs to be added to a DS-3. A very similar system was developed in Europe, except instead of using DS-n signals, the European PDH used European-n (E-n) signals. Table 2.1 shows the data rates for the North American and European PDHs.

**TABLE 2.1. Data rates for American and European Plesiochronous Digital Hierarchies.**

<b>Signal Level</b>	<b>Data Rate (Mbits/s)</b>	<b>Signal Level</b>	<b>Data Rate (Mbits/s)</b>
<b>DS-0</b>	0.064	<b>E-0</b>	0.064
<b>DS-1</b>	1.544	<b>E-1</b>	2.048
<b>DS-2</b>	6.312	<b>E-2</b>	8.5448
<b>DS-3</b>	44.736	<b>E-3</b>	34.368
<b>DS-4</b>	274.176	<b>E-4</b>	139.264
<b>DS-5</b>	400.352	<b>E-5</b>	565.148

The SONET standard uses the synchronous transport signal 1 (STS-1) signal as the base standard. Each STS-1 frame is organized into a 90 column by 9 row byte structure and has a frame rate is 8000 frames/second. At 810 bytes per frame, the total STS-1 bit rate is 51.840 Mbits/s. The first 3 columns of each frame is allocated to the transport overhead which is used for monitoring, protection switching, fault section-alization, signal framing, line identification, performance monitoring as well as voice and data channels used for maintenance and provisioning. The latter 87 columns

make up the synchronous payload envelope (SPE), the first column of which contains path specific functions such as end-to-end performance monitoring and path identification. The latter 86 columns of the SPE contain payload signals.

An STS-1 SPE can carry a single DS-3 signal. The SPE can also be divided into virtual tributary (VT) groups. These groups are designed to hold sets of virtual tributary signals and come in varying sizes. They are intended to provide backwards compatibility to lower rate signals, such as DS1 and DS2. Higher order STS signals are created by byte interleaving or concatenating STS-1 signals, resulting in STS-n or STS-nc signals respectively. The standard multiples used are STS-3, STS-12, STS-48 and STS-192 as well as STS-768. These digital signals are converted into corresponding OC-n signals used by SONET optical equipment. The OC signals correspond to the STS data rates after electrical to optical conversion. For example, an STS-24 signal is converted to an OC-24 signal for transmission.

The international version of SONET is called the synchronous digital hierarchy (SDH) [20], which is similar in functionality to SONET, but adheres to European standard digital rates. The SDH standardizes multiples of the OC-3 SONET signal and these levels are defined as Synchronous Transport Modules (STM). Table 2.2 shows the standard SONET and SDH data rates.

**TABLE 2.2. Standard SONET/SDH data rates.**

<b>SONET Signal Level</b>	<b>Optical Signal</b>	<b>SDH Equivalent</b>	<b>Data Rate (Mbits/s)</b>
STS-1	OC-1	-	51.84
STS-3	OC-3	STM-1	155.25
STS-12	OC-12	STM-4	622.08
STS-48	OC-48	STM-16	2488.32
STS-192	OC-192	STM-64	9953.28
STS-768	OC-768	STM-256	39813.12

## 2.8.2 WDM/DWDM/CWDM

Wavelength division multiplexing (WDM) [15] is an optical transmission technology used to send several different carrier wavelengths down the same fiber simultaneously. The carrier signal operating frequencies fall into 1310nm and 1550nm ranges. These regions are used because in these ranges the fiber attenuates the signal the least. Coarse wavelength division multiplexing (CWDM) is a version of this technology that establishes 2 to 4 wavelength channels on a single fiber and is used to increase the total capacity in the network without installing additional fiber. Dense wavelength division multiplexing (DWDM) allows for hundreds of closely spaced carrier wavelengths to be sent simultaneously over a single fiber. The DWDM C-band defines a grid of 80 wavelengths starting at 1528.77nm that are spaced 50 GHz apart. These wavelengths are precisely spaced and their frequency stability and power levels are closely monitored in order to minimize interference.

## **2.9 Network Survivability**

As discussed briefly in Chapter 1, modern transport networks are complicated systems, the individual components of which are subject to failure. These failures have an effect on the overall performance of the transport network, but the effect is generally not noticed by the users of the network because of the survivability measures in place. Network survivability, referring to the ability of the network remain continuously operational despite individual component failures, is the driving idea in the field of network planning.

There are two main areas where survivability can be built into the network. The first is at the nodal equipment level residing in the protected CO buildings. Redundant power feeds, supplies and converters, redundant high speed optical line cards, detectors, lasers, as well as backup computer systems and memory: all this contained in secure locations with regular maintenance. All of these factors contribute to a high operational level availability at the nodes of the network. Availability refers to the probability that a system will be found in operational condition at an arbitrary

time. This is not to be confused with reliability which is the probability that a system remains operational for the duration of its mission. The previously mentioned equipment and maintenance deals with failures occurring within the system itself and a large portion of the network cost comes from these facilities.

Survivability is also vital at the interconnections between the nodes. The fiber optic cable connecting nodal equipment is particularly vulnerable as much of it is suspended above ground, buried shallowly, or simply resting on river or lake beds. The reason that every meter of cable is not deep below ground is because it is too cost prohibitive to do so. While some protection is provided by strong sheath materials surrounding the optic fibers, cable cuts are unavoidable and are the most common type of failure in a transport network. What to do in the event of such failures and how to best prepare for when they do happen are questions that the field of network planning attempts to find answers to. As will be seen in the following subsections there exist several techniques for protecting the network from optical fiber cuts. These vary in speed of recovery, complexity, ease of implementation, and capacity/cost efficiency. Often there is no “best” technique to use and many factors must be considered.

### **2.9.1 Protection and Restoration**

The most basic concept of survivability in transport networking follows that for any working path to survive a single failure, an alternate path dedicated to that demand that is not affected by the failure must be found in the network. The survivability schemes covered in Section 2.10 and Section 2.11 all share this goal. Based on how these schemes are implemented they can be placed on a scale between protection and restoration which categorizes how these schemes react to failure. Pure protection schemes are ones where the spare capacity is fully connected into protection structures and every backup path is known in advance, tested and can be accessed as soon as a failure occurs. The system is in a ready state and minimal signaling is required between nodes in order for the network to recover from failure. Pure restoration

schemes, by contrast, are ones where the spare capacity is not connected before failure and backup paths are found in real-time when a failure occurs. Restoration schemes tend to be slower than protection schemes because of the time required for restoration schemes to determine the replacement paths for affected working paths or path segments. Network-wide signalling is required for failure notification and to ensure that spare capacity is correctly cross-connected. Schemes where the backup paths and cross-connection tables are known in advance but where spare capacity remains unconnected until failure are referred to as pre-planned restoration schemes. The survivability architectures that act at the physical layer, described in Section 2.10, are categorized as protection schemes where as the architectures that act at the logical layer, described in Section 2.11 can be either pure or pre-planned restoration or protection.

## **2.9.2 Redundancy**

There are two important metrics that are examined when designing a transport network: the total cost and redundancy. Total cost allows different designs to be compared to one another and can represent overall installation, operation and/or maintenance costs. It takes into account both the resources required to route the working demands as well as the costs associated with the mechanism implemented for network survivability. In this thesis the total cost will generally represent the total equipment cost of the network. Another related metric used in this thesis is total capacity. Total capacity is essentially the sum of working and spare capacities used and represents the total number of capacity channels needed on every span to implement the design. Redundancy is the measure of how elegant or lean a particular design is. It basically answers the question of what is the additional cost required, relative to the cost of routing all working demands without protection, to reach a desired level of survivability. The desired level of survivability can be, for example, 100% protection against single span, node or dual span failures.

Redundancy can be calculated based on either cost or capacity. Cost redundancy can be calculated using equation (2.1) where  $S$  is the set of all spans in the network,  $w_i$  and  $s_i$  represent the working and spare capacity quantities and  $c_i$  represents the cost of capacity on each span  $i$  or the geographic distance of that span. For this reason cost redundancy can also be referred to as geographical redundancy. Capacity, or logical redundancy can be calculated using equation (2.1) by setting every  $c_i$  coefficient to 1. In this case redundancy is simply the pure ratio of working to spare capacity without taking any distance or equipment costs into account. The type of redundancy used to evaluate a network design is based on whether the total cost or total capacity was minimized as part of the design process.

$$R = \left( \sum_{i \in S} c_i \cdot s_i \right) / \left( \sum_{i \in S} c_i \cdot w_i \right) \quad (2.1)$$

Strictly speaking, equation (2.1) should only be used when the working capacity uses shortest or fewest hops routing. This is generally the case where the network design was obtained using a spare capacity placement (SCP) method. In SCP, the working capacity is routed based on lowest cost or fewest hops independently of spare capacity. In the case when working capacity is not lowest cost or fewest hops routed, standard redundancy (equation (2.2)) is used. This case arises in joint capacity placement (JCP) designs where the working routing is not restricted to being lowest cost or fewest hops and working and spare capacity placement is determined simultaneously. Equation (2.2) reuses the variables and coefficients introduced in equation (2.1). A new variable  $w_{min,i}$  represents the working capacity used in the minimum cost or fewest hops working routing where as  $w_i$  represents the actual working capacity used. The first term in the numerator of equation (2.2) represents the total cost of the network while the second term in the numerator as well as the denominator represent the working capacity cost that would be required if lowest cost working routing was used.

Basically this equation gives the ratio between the spare capacity in addition to the overhead working capacity cost to the minimum working capacity cost possible. It states that any overhead working capacity cost that is used over that of the shortest working routing should be counted together with the spare capacity and placed in the numerator. At first glance, it would appear that deviating from lowest cost routing would actually increase the total cost of the network and adding overhead working capacity to the spare capacity would increase redundancy; the two metrics that network designers try to minimize. However, in JCP designs, deviating from lowest cost working routing allows for significant reductions in spare capacity cost. So much so that despite the slight working capacity cost increase, the reduction in spare capacity cost (relative to the minimum cost SCP design) is significant enough that the total cost of the network is decreased. Note that if standard redundancy is applied to an SCP design, equation (2.2) can be reduced to equation (2.1).

$$R_{std} = \frac{\left( \sum_{i \in S} c_i \cdot (w_i + s_i) \right) - \left( \sum_{i \in S} c_i \cdot w_{min, i} \right)}{\left( \sum_{i \in S} c_i \cdot w_{min, i} \right)} \quad (2.2)$$

## 2.10 Survivability at the Physical Layer

The following sections introduce survivability techniques used in transport networking which can be categorized into two types: physical and logical. At the physical level, the transmission network is not aware or concerned with the service-specific routing that the upper level traffic takes. What the network sees at this level are the aggregations of data and traffic flows going from node to node. The survivability techniques at the physical level operate by essentially rerouting the affected traffic from a failed to a backup fiber. These survivability techniques are generally categorized as protection, and cannot be easily reconfigured. Once installed and tested, these systems are discrete simple structures in the transport network, providing pre-connected, end-

node activated replacement fiber routes for failed spans. Physical level survivability schemes do not require network-wide signaling at the time of failure and exhibit very fast recovery times.

### **2.10.1 Automatic Protection Switching**

Automatic Protection Switching (APS) is a general name for the most basic and historically prevalent network survivability technique. The fastest and most straight forward version of this technique is called 1+1 APS [21]. It involves sending the same signal from the origin node to the destination node over two different channels routed over the same spans using head-end bridging at the transmitter end. These two channels are designated as working and protection channels and it is up to the receiver to detect the failure of the working path and to switch over to the backup in a process called tail-end transfer. If the two channels are over physically diverse routes, the arrangement is referred to as 1+1 APS DP, where DP stands for diverse protection. In this arrangement protection from a single failure on a fiber is guaranteed. APS 1+1 is the fastest architecture available because the only action required for restoration is for the receiver to switch over to the backup channel. This action takes place without having to notify the transmitter and in an optical transmission network the receiver makes the switch as soon as a loss of light is detected. The downside of 1+1 APS is the high cost that comes with implementing this survivability scheme. This is the only network protection technique that does not allow for any sharing of the backup paths. Virtually every component of the network has to be doubled, and implementing 1+1 APS is essentially like building two near identical networks, one on top of the other.

1:1 APS is an arrangement where there instead of sending the same signal down two different channels, during regular operation the signal is only sent down the working channel. The backup channel is reserved in case of a failure and can either be idle or containing low priority traffic that can be pre-empted in case of a failure. This technique is slower than 1+1 APS because the receiver must signal the transmitter

when a failure occurs to establish the head-end bridge so that recovery may take place. The benefit of this technique is that it does not lock the backup path in a dedicated arrangement and frees up this capacity for other uses such as maintenance as well as low priority traffic.

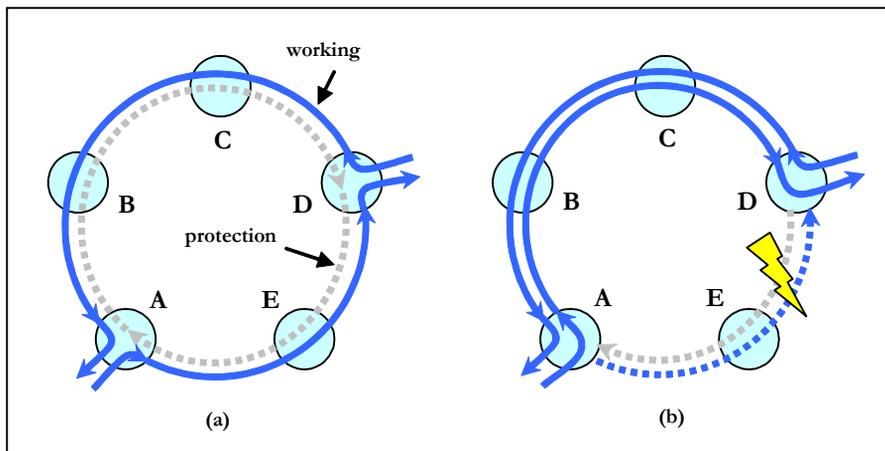
1:N APS is an extension of 1:1 APS where  $N > 1$  and the standby backup channel is shared among  $N$  working channels for protection. When a failure occurs, the receiver checks to see if the backup channel is in use, and if it is not, a signal is sent to the transmitter to establish the head-end bridge of the failed channel into the backup channel. APS can further be extended to  $k:N$  APS ( $k > 1$ ,  $N > 1$ ) where  $k$  protection channels are allocated to protect  $N$  working channels. This method is the slowest APS method since additional logic must be employed to manage resources during failures. Generally speaking, 1:N APS and  $k:N$  APS are used to protect the network against internal failures and do not protect against single fiber cuts. The working and protection channels are routed together and protection is provided against line card or other internal failures that would affect only one channel and not the whole fiber optic cable. 1+1 APS and 1:1 APS arrangements are used for fast recovery of external failures, such as fiber cuts, as long as the working and standby protection channels are diversely routed.

### **2.10.2 Self-Healing Rings**

After APS, self-healing rings are the next simplest survivability architecture. Rings are cyclical structures where spare and working capacity is equally provisioned on every span the ring crosses. Unlike APS 1+1 DP where the entire path is protected end-to-end by a single structure, it is possible for a single working path to be segment-wise protected by several rings. In this case, each of these rings would only be responsible for protecting the portion of the working path that crosses it and can do so at very high speeds. One of the biggest advantages of rings over APS is the use of ADMs as nodal devices instead of fiber being physically patched through at the nodes. This

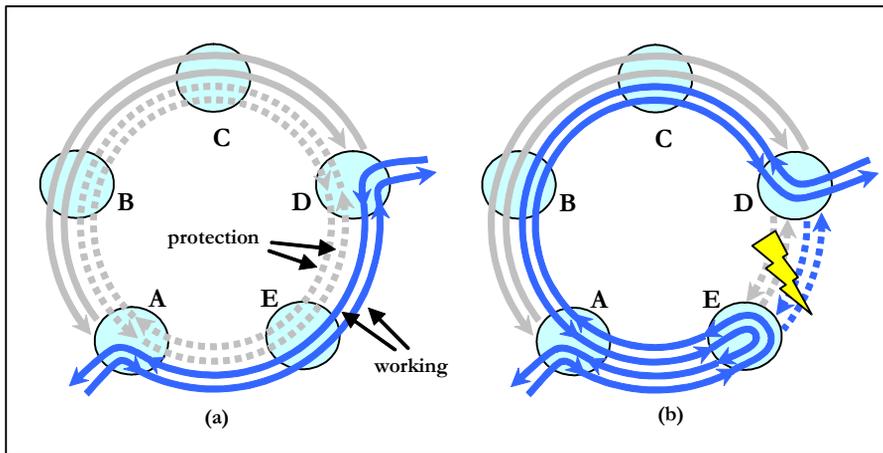
allows network operators to add or drop channels and essentially reconfigure the ring as needed as long as the capacity of the ring is not exceeded.

The two main types of rings are Unidirectional Path Switched Rings (UPSR) [22] and Bidirectional Line Switched Rings (BLSR) [23]. A UPSR is a structure that requires only 2 fibers and is illustrated in Figure 2.7 in both a working and a failed state. In the working state, a bidirectional demand is routed between nodes A and D on opposite segments of the ring in a counter-clockwise direction on working fiber. Protection fiber is shown going in the clockwise direction. Just like in APS 1+1, the working signal is sent down the protection fiber simultaneously. In the event of a failure, the receivers at the nodes used to access the ring (A and D in this example) perform a tail-end transfer to select the backup signal, re-establishing the failed connection. One notable property of UPSRs is that any demand routed through the ring must traverse every span of the ring. This means that the total (working and protection) capacity of the ring must equal to the sum of all the demands protected by the ring. UPSRs are widely deployed using the SONET framework. The WDM-based equivalent of UPSRs are Optical Path Protection Rings (OPPRs) [24].



**FIGURE 2.7. Illustration of UPSR operation (a) before failure and (b) after failure.**

Where a UPSR is similar to several 1+1 APS systems over the same spans, a BLSR is a cyclical structure that can be viewed as overlaid 1:N APS systems where the protection capacity is shared among multiple different working sections. Figure 2.8 illustrates a 4-fiber BLSR in a working and failed state. In a 4-fiber BLSR, there are two sets of fibers; one set of dedicated working fibers and one set of dedicated protection fibers. The sets of working and protection fibers allow for bidirectional transmission on any span of the ring. This allows for a demand to be routed between two nodes of a BLSR using only a segment of the ring (typically the shortest) as opposed to being forced to use the entire ring as was the case for UPSRs. An example of this is illustrated in Figure 2.8a, where the demand shown is routed between node A and D using spans A-E and E-D. The rest of the spans (A-B, B-C, C-D) remain unused and can be used to route other demands. All the demands protected by a BLSR share the spare capacity on it. Because of this, BLSRs only need as much capacity as the largest total of the demand flow crossing any span of the ring. In the event of a failure (Figure 2.8b), the nodes adjacent to the failed span loop-back the entire working traffic into the protection fibers. Overhead signaling is required to ensure that the appropriate nodes perform the switching action. The 2-fiber variant of the BLSR works exactly like the 4-fiber version except that in the 2-fiber case, the channels on each fiber are split into the working and spare categories instead of using whole fibers for each, as was the case in 4-fiber BLSRs. The WDM-based equivalent of BLSRs are Optical Shared Protection Rings (OSPR) [25].



**FIGURE 2.8. Illustration of BLSR operation (a) before failure and (b) after failure.**

The main benefits of ring systems is that they have very fast restoration times. The protection path is known well in advance of failure and restoration action only involves the nodes adjacent to the failed span. Ring systems are simple, very easy to operate and can be implemented using low cost ADM equipment. One of the main down-sides, however, is that ring systems have very poor capacity efficiency. UPSR networks have very high capacity requirements as any demand routed on a ring uses the entire ring for transmission and the UPSR does not allow for any protection capacity sharing. BLSRs are slightly more efficient than UPSRs because they allow different demands routed on the same ring to share protection capacity. However, this slight increase in efficiency is offset by the fact that it is difficult to find working routing that maximally utilizes every BLSR in a multi-ring network. Because of this a large amount of transmission capacity ends up being stranded and cannot be used. For rings of very large size the distance traversed by the protection path can be extremely long, especially for rings connecting different continents. Furthermore, efficient multiple ring network design is a complex problem, and generating optimal solutions for even the medium sized networks require extensive computational resources. Because of this heuristics are heavily relied upon for ring network design. Rings are not very

amendable to manual redesigns and adding rings to a network as demand grows often results in inefficient designs. Despite these shortcomings, rings (and APS systems) are commonly deployed as they were the most widely standardized and commercially available survivability solutions in the 1990s.

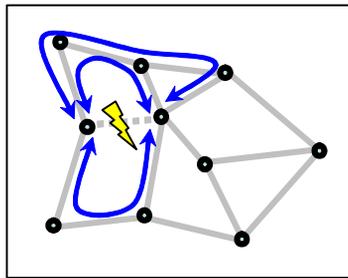
## **2.11 Survivability at the Logical Layer**

Logical layer survivability techniques view the individual channels between adjacent nodes as generic capacity that can be used for demand routing and protecting these demands against failure. Working capacity can be arranged into shortest/least hops routes. This is in contrast to ring systems, for example, where working routing is forced to follow the spans crossed by the ring. Spare capacity can be unorganized and can exist as a pool of capacity to be used when failure occurs, as is the case in restoration-logical-layer survivability techniques. Or it can be organized into identifiable protection structures such as trails, cycles or trees, which can be either pre-connected or not, as is the case for protection and pre-planned restoration survivability techniques, respectively. The following sections introduce the various logical survivability techniques that have been introduced for transport networking. The goal of these sections is to acquaint the reader with the basic concepts related to each particular architecture. More detailed discussions on these architectures may be found in [9] as well as the references provided.

### **2.11.1 Span Restoration**

Span restoration (SR) [26] is a survivable mesh networking technique where only the nodes adjacent to the failure are responsible for restoring the demands affected by it. This is a variant of mesh networking where restoration is localized and is also referred to as link protection or line restoration. Mesh networking refers to architectures where spare capacity can be arbitrarily arranged into backup paths, utilizing the mesh-like physical topology of the network. Prior to failure, the spare capac-

ity is unused or is used to carry low priority traffic that can be pre-empted in case of failure. When the a span failure occurs, the adjacent nodes use spare capacity to restore the connectivity between the two nodes for all the affected demands. This technique works by providing logical detours in the form of replacement paths connecting the nodes adjacent to the failed span. An example of this is illustrated by Figure 2.9 where a failure of a span affecting three units of working capacity is restored using three replacement paths (dark arrowed lines) originating and terminating at the nodes adjacent to the failure. These paths are effectively spliced into the working path and replace the failed channel. The main design objective for SR is to ensure that for any single span failure, there is enough spare capacity in the network to re-connect the affected working capacity. The efficiency of this method comes from replacement paths from different span failures sharing spare capacity.



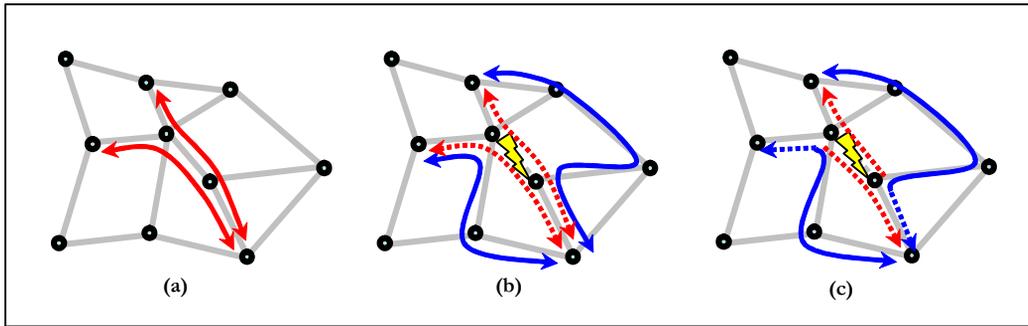
**FIGURE 2.9. Span restoration example showing three replacement paths used to restore three units of working capacity affected by failure.**

### 2.11.2 Path Restoration

Path restoration (PR) [27], [28] is a mesh survivability technique that is similar to SR in operation and concept. Both use replacement paths built out of shared spare capacity in case of a failure and the paths are established only after a failure occurs. The main difference between SR and PR is that in SR the replacement paths are established connecting the adjacent nodes of the failed span in contrast to PR

where the replacement paths originate and terminate at the affected demand's end-nodes. PR is a variant of mesh networking where the restoration is end-to-end. The actual restoration paths used to restore a particular demand can vary based on where in the network the failure occurs. When a single span failure affects a working path crossing several spans, the unaffected working capacity channels of the failed working path are referred to as stubs. These stubs can be added to the pool of spare capacity in an action called "stub release" and can then be used in addition to spare capacity to generate replacement paths for failed demands. PR with stub release has important theoretical significance because it represents the most efficient survivability technique and can be used as a lower efficiency bound for all other architectures. It is possible to achieve an even lower efficiency bound than PR with stub release by reorganizing both working and spare capacity after a failure. This type of technique is not used because of the implications of doing so from the point of view of network reliability and availability. In general, any protection architecture that forces working paths unaffected by failure to be terminated and re-routed for the sake of lower capacity use is viewed as not acceptable in transport networking where the aim is to shield working paths from failure as much as possible.

An example of PR in an event of a single failure scenario is illustrated in Figure 2.10. Two operational working paths are shown using light arrowed lines in Figure 2.10(a). Figure 2.10(b) illustrates single span failure recovery action taken when PR without stub release is used. The replacement paths taken to recover the failed demands are shown using dark arrowed lines. Note that the replacement paths do not have to be disjoint from the working paths (except for the span affected by the failure) if doing so reduces the total capacity of the design. Figure 2.10(c) illustrates single span failure recovery when PR with stub release is used. In the case shown, two units of working capacity are reused. This results in two fewer units of spare capacity required to protect the same amount of working capacity, relative to PR without stub release.



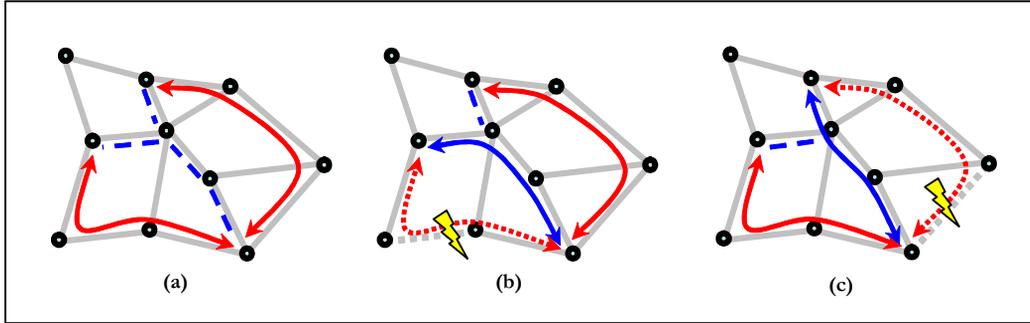
**FIGURE 2.10. Path restoration single failure example: (a) Two working paths prior to failure. (b) PR without stub release. (c) PR with stub release.**

### 2.11.3 Shared Backup Path Protection

Shared backup path protection (SBPP) [29] is a pre-planned protection architecture that combines aspects of PR and 1:1 APS DP. Under SBPP every working path is provided with a single end-to-end protection path that is disjoint from it (much like 1:1 APS DP). This allows for the same protection path to be used regardless of which span crossed by the working path is affected by failure. This property of SBPP is referred to as failure independence while it can be said that PR is a failure dependant architecture. Similarly to PR, under SBPP the spare capacity is shared among protection paths used in different failure scenarios. Another way of looking at it is that sets of span disjoint working paths may have their backup paths share spare capacity. This is because no two disjoint working paths will ever be affected by the same span failure and will therefore never contend for spare capacity when only single span failures are considered. This concept can further be extended to protect against node failures by making sure that the working paths are mutually span and node disjoint.

The concept of SBPP illustrated in Figure 2.11. Two disjoint working paths, represented by solid arrowed lines, as well as unconnected spare capacity, represented by the long dashed lines are shown in Figure 2.11(a). Two failure scenarios affecting

each working path are shown in Figure 2.11(b) and Figure 2.11(c). The respective backup path used to restore each working path is shown as well, represented by solid dark arrowed lines. Note that because the two working paths were disjoint, their backup paths were allowed to share capacity on two spans.

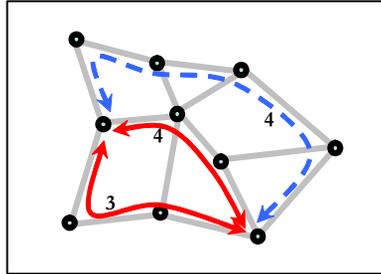


**FIGURE 2.11. SBPP example. (a) Two working paths and unconnected spare capacity. (b,c) Two single span failure scenarios and the respective backup paths used.**

### 2.11.4 Demand-Wise Shared Protection

Demand-wise shared protection (DSP) [30], [31] is a recently developed path protecting architecture where the spare capacity is shared between different paths of individual demand bundles. Under DSP, the working and protection paths of each demand bundle are routed in integral amounts over spatially diverse routes. Figure 2.12 illustrates an example of a DSP 3-way split arrangement where three diverse routes are used to realize a demand request of 7 units. The solid arrowed lines represent the working routes carrying 3 and 4 working paths respectively. The dashed arrowed line represents a protection route carrying 4 protection paths. The number of protection paths provisioned must equal the number of working paths that can be affected by a single failure at the same time. If either of the two working routes fail, the working paths on these routes can be recovered using the protection paths provided. Some of the desirable properties of DSP include the fact that, similarly to APS,

only the end-nodes are responsible for switching actions when a failure occurs. DSP can be equal to or better than 1+1 APS in terms of capacity use, can theoretically approach redundancies of optimal SBPP solutions [31] and can be implemented using simple networking equipment.



**FIGURE 2.12. DSP 3-way split example for a 7 unit demand bundle.**

DSP can be efficient in very specific situations where three or more mutually disjoint routes of approximately the same length exist between a pair of nodes and where there exists a large enough demand requirement to utilize all of the available routes. This ideal situation is very rarely realized in DSP designs. Often, the third or higher order routes are far too long to yield significant cost savings relative to using just a 2-way split arrangement. Note that in the example illustrated in Figure 2.12, a total of 21 units of working capacity are protected using 20 units of spare capacity. In contrast, in an APS equivalent 2-way split arrangement (where only the shortest two routes are used: one for working and the other for protection paths) uses a total of 21 units of working capacity and 21 units of spare capacity. In all, an upgrade from a 2-way to a 3-way split only saves only one unit of capacity. This corresponds to only a 2.4% improvement, relative to APS. Because of this effect, in addition to the fact that spare capacity is not shared between different demand bundles, general network DSP solutions are only slightly better than APS in terms of capacity [32].

### 2.11.5 Pre-Cross-Connected Trails

Pre-cross-connected trails (PXTs) [33], [34] is a path protecting pre-connected architecture where spare capacity is organized into non-cyclical degree 2 (except for the end-nodes) structures called trails. These trails do not contain any branch points and are allowed to cross the same nodes and spans several times in the general case, if it is optimal to do so. PXTs that do not cross the same span more than once are referred to as simple PXTs. A working path is eligible for end-to-end protection by a PXT if the PXT crosses its end-nodes and the protection path provided by the PXT and the working path are mutually disjoint. Spare capacity on a single PXT can be shared between working paths that cannot be simultaneously affected by a single failure. Another way of saying this is that sets of mutually disjoint working paths are eligible to be protected by the same PXT. It is possible for two non-disjoint working paths to be protected by the same PXT but only if the protection paths provided to the working paths by the PXT are mutually disjoint.

Note that PXTs are similar to SBPP in the respect that every working path is assigned a single protection path that is activated regardless of where the failure affects the working path. However, under PXTs the spare capacity is organized into trails that are pre-connected and can be tested before failure occurs, where as in SBPP the spare capacity remains unconnected and is only cross-connected after failure. An example PXT configuration is shown in Figure 2.13 where a set of 3 working paths (solid arrowed lines) is protected by a single PXT (dashed arrowed line). In the case of a failure, the end-nodes of the affected working path initiate a switching action that breaks into the PXT, restoring the failed path.

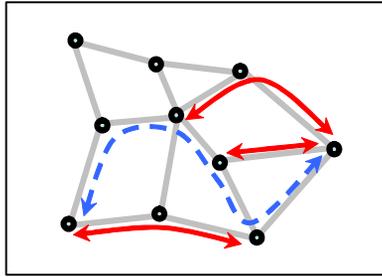


FIGURE 2.13. Example of a single PXT protecting 3 working paths.

### 2.11.6 $p$ -Cycles

$p$ -Cycles (also referred to as span-protecting  $p$ -cycles) [35], [36], [37] are cyclical, degree 2, pre-connected structures that provide protection for channels on spans whose adjacent nodes are crossed by the cycle. An example of a unit capacity  $p$ -cycle in a pre-failure state is illustrated by the connected dark line in Figure 2.14(a). They offer restoration speeds comparable to that of ring networks while also providing mesh-like efficiency. The high speeds associated with  $p$ -cycles come from the fact that the spare capacity is pre-connected and that every restoration action is pre-planned so that the only actions required after a failure is that of failure identification and accessing the  $p$ -cycle.  $p$ -Cycles offer protection against two types of failures: on-cycle and straddling.

On-cycle failure protection is provided to the working channels on spans crossed by the cycle. In an event of an on-cycle span failure, the end-nodes of the span perform a restoration action that re-routes the failed working channels into the surviving segment of the cycle. This is illustrated in Figure 2.14(b) where the arrowed solid line represents the replacement path taken to restore the unit channel on the failed span. This operation is very similar to the way that BLSRs provide protection against span failure. Also similarly to rings, every span crossed by the cycle can be protected in this way. So far, only taking into account on-cycle failures, this cycle is 100% redundant: there are 10 units of working capacity protected by 10 units of spare capacity.

Straddler failure protection is provided to spans that are not crossed by the cycle but whose adjacent nodes are. Failures affecting straddler spans do not affect the cycle itself. This allows the cycle to be split into two segments originating and terminating at the nodes adjacent to the failure. These two segments provide two replacement paths that can be used to restore two units of affected working capacity on the failed span. Figure 2.14(c) shows an example of a straddling span failure where the cycle is broken into at the end-nodes of the failed span, providing two replacement paths, represented by the arrowed lines. Straddler protection is the key to the high mesh-like efficiency provided by  $p$ -cycles. The cycle in Figure 2.14(a) has 5 straddling spans and can protect 10 units of working capacity on this spans. This is in addition to the 10 units of on-cycle working capacity already protected from the example above, when straddlers were not considered. In total this cycle can protect 20 units of working capacity using only 10 units of spare capacity which equates to 50% redundancy. This efficiency is comparable to that of SBPP, path restoration and span restoration and is significantly more efficient than what is possible using ring and APS systems.  $p$ -Cycles have another major advantage over rings. In addition to high efficiency, under  $p$ -cycles working capacity can be routed using the shortest/fewest hops routing. This is in contrast to working paths being restricted to follow the spans crossed by rings in BLSR and UPSR systems.

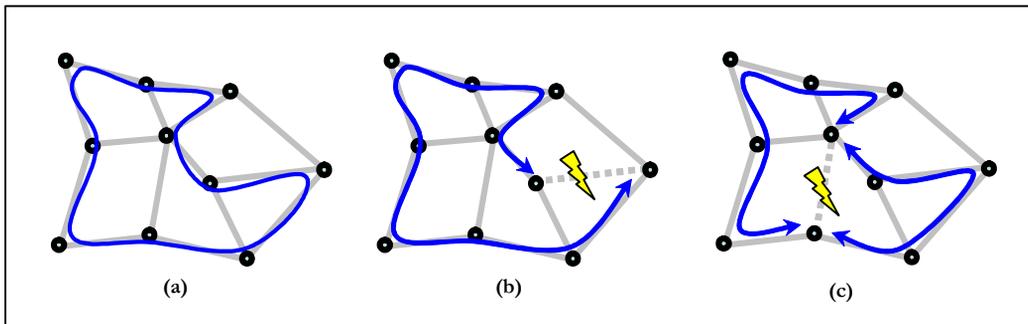


FIGURE 2.14. Example of (a) a  $p$ -cycle and the action taken after (b) an on-cycle and (c) straddling span failure.

### **2.11.7 FIPP $p$ -Cycles**

Failure independent path protecting (FIPP)  $p$ -cycles [38], [39], [40] are an extension of span-protecting  $p$ -cycles to path protection. Similarly to span-protecting  $p$ -cycles, FIPP  $p$ -cycles are cyclical, degree 2 structures that are capable of offering on-cycle and straddler protection. The main difference between the two is that where span-protecting  $p$ -cycles only protect individual spans, FIPP  $p$ -cycles protect entire paths working paths, end-to-end, as long as the end-nodes of the working paths cross the same nodes as the cycle. FIPP  $p$ -cycles are the main focus of this thesis and for this reason, the details regarding this architecture, including a thorough literature review, are deferred to Chapter 3.

### **2.11.8 GPP $p$ -Cycles**

Generalized path protecting (GPP)  $p$ -cycles [41] are a recently proposed member of the path protecting  $p$ -cycle family. Essentially GPP  $p$ -cycles are operationally similar to FIPP  $p$ -cycles except that under GPP the failure independence constraint is lifted. This means that working paths protected under GPP can be protected using different configurations depending on where the working path is affected by failure. This is in contrast to FIPP  $p$ -cycles, where each working path can only be protected using a single configuration to ensure that the same switching action takes place regardless of the location of the failure. Because of how closely related this architecture is to FIPP, further details regarding GPP  $p$ -cycles are deferred to Chapter 3.

### **2.11.9 Flow $p$ -Cycles**

Flow  $p$ -Cycles [42], [43] are an extension of the span-protecting  $p$ -cycle concept with the main difference being that they protect path segments instead of individual spans. The path segments protected by flow  $p$ -cycles may be of any size, be it a single span, the entire working path or just a segment of it. For a path segment to be

eligible for protection, its end-nodes must fall on the nodes crossed by the cycle. Similarly to span-protecting  $p$ -cycles, flow  $p$ -cycles allow for on-cycle and straddler failure protection. Any working path segment protected by a cycle that is span disjoint from the cycle is eligible for straddler failure protection. Any segment that crosses on at least one span also crossed by the cycle is eligible for on-cycle protection. The general rule is that a cycle may protect a set of path segments if the path segments are mutually disjoint from one another and cannot fail simultaneously. If the path segments are not mutually disjoint they can still be protected by the cycle if their replacement paths are. In other words, if two segments do fail simultaneously (because they were not mutually disjoint), they can be protected by the same cycle if they do not contend for the same spare capacity on that cycle.

Flow  $p$ -cycles provide the most general solution out of all  $p$ -cycle architectures. Both FIPP  $p$ -cycle and span-protecting  $p$ -cycle solutions can be flow  $p$ -cycle solutions. FIPP and span-protecting  $p$ -cycles fall on opposite sides of the span to path protection spectrum. Flow  $p$ -cycles are able to cover the middle ground as well as the extremes if it is optimal to do so. Flow  $p$ -cycles significantly extend the ability of span-protecting  $p$ -cycles to protect against node failures. This was previously not possible with span-protecting  $p$ -cycles because node failures affect not only the working path but also the cycle access point required for the restoration action to take place.

## 2.12 Optimization and Linear Programming

The following sections cover the details regarding how efficient transport network designs can be obtained. Significant attention is given to linear programming (LP), which is the main method used for obtaining network designs in this thesis. It lends itself gracefully for use in transmission networking problems, where the relevant elements in the network as well their interrelations can be efficiently described by modeling these as a linear program (LP). Before proceeding to introduce LPs in detail, the following section provides a brief overview of relevant set theory concepts.

## 2.12.1 Set Theory and Notation

The *set* is one of the most fundamental concepts in mathematics [44]. It is an unordered, non-repeating collection of objects (called *elements*) that are related to one another by the inclusion in the set. A set itself can also be an object and is completely defined by the elements contained within. Using common notation, a set  $S$ , containing elements  $a$ ,  $b$  and  $c$  can be written as  $S = \{a, b, c\}$ . The order in which the elements are written when defining a set does not matter, meaning that the above is equivalent to  $S = \{b, c, a\}$  or  $S = \{b, a, c\}$ , for example. It is also possible to have a set that contains no elements at all, referred to as a null set. Using common notation a null or empty set is defined as  $P = \{ \}$ , as well as  $P = \emptyset$ . The cardinality of a set is defined as the number of elements contained within the set, as in  $|S| = 3$  and  $|P| = 0$ . The character  $\in$  is used to describe whether a particular element is in a particular set or not. For example  $a \in S$  states that element  $a$  belongs to set  $S$ . Conversely,  $\notin$  is used to describe when an element does not belong to a set as in  $d \notin S$  which states that element  $d$  does not belong to set  $S$ . To refer to all the elements of a set, without explicitly naming all the elements, the universal quantifier  $\forall$ , also meaning “for all” or “for each”, is used. For example,  $\forall x \in S$  can be read as ‘for each element  $x$  in set  $S$ ’. The existential qualifier  $\exists$  is used to indicate a test to see if a specified object is a member of a given set and is usually read as “if exists” and  $|$  means “such that”. For example, these qualifiers can be used in an expression such as  $\langle \forall x \in S | \exists y \in T, x \leq y \rangle$  which can be read as follows: For all  $x$  within set  $S$  such that there exists an element  $y$  within set  $T$  where  $x$  is less than or equal to  $y$ .

If two sets  $S$  and  $T$  contain exactly the same elements then they are said to be equal. This can be represented in a statement such  $S = T$ . If set  $S'$  is made up completely of the elements from set  $S$  it is said to be a subset of  $S$ . Using mathematical notation, this is the same as writing  $S' \subseteq S$  or  $S \supseteq S'$  where  $\subseteq$  is referred to as

inclusion. Conversely, it can also be said that  $S$  is a superset of  $S'$ . Under inclusion it is permissible for set  $S'$  to be made up of every element  $S$ , in other words inclusion is not violated if  $S = S'$ . A subset  $S'$  is a proper subset if it is made up of elements from its superset  $S$ , but  $S \neq S'$ . Using mathematical notation this can be written as  $S' \subset S$  or  $S \supset S'$  where  $\subset$  is referred to as strict inclusion.

A union of two sets  $S = \{a, b, c\}$  and  $T = \{b, c, d\}$  is defined as the set of all elements that appear in either (or both) sets, and can be written as  $S \cup T = \{a, b, c, d\}$ . The intersection of two sets is defined as the set of all elements that appear in both sets. Using the previous sets as an example, this can be written as  $S \cap T = \{b, c\}$ . The difference between two sets is defined as the set of elements from belonging to the set being subtracted from that do not appear in the set that is being subtracted. In this case the order in which the sets appear in the expression matters and this can be written as  $S - T = \{a\}$  or as  $T - S = \{d\}$ . A symmetric difference between two sets is defined as the set of elements that appear in either set, but not in both. This can be written as  $S \Delta T = \{a, d\}$ .

### **2.12.2 Linear and Integer Linear Programming**

The LP techniques and terminology extensively used in this thesis are borrowed from the field of Operations Research (OR) which is a field dedicated to the use of mathematical modeling and algorithms to arrive at optimal or near optimal solutions to complex problems. This thesis, however, focuses on the application of these techniques to practical engineering problems in transport networking, not on the techniques themselves. This section is intended to provide a basic, user level understanding of OR methods and terminology to aid the reader in understanding the models presented in this thesis. Detailed information about the techniques and mathematical algorithms for solving LPs can be found in the following references:

[45], [46], [47]. Additionally, a detailed survey of OR methods relevant to network design can also be found in [9].

Optimization, or mathematical programming, is the study of problems in which the main goal is to minimize or maximize a mathematical equation (also referred to as the objective function) that relates several decision variables, subject to variable constraint equations (referred to as constraints). Note that in this context, the term programming does not relate to the current common use of the word: the programming of a computer. Instead, here the term programming, relates to the concept of scheduling or planning. The problems encountered throughout this thesis can be categorized as combinatorial optimization problems. In these types of problems the set of feasible solutions (solutions that satisfy all of the problem's constraints) is discrete or can be reduced to a discrete one. These problems involve selecting combinations of smaller discrete elements to form solutions and the number of feasible solutions increases rapidly with problem size. It is relatively easy to find a feasible solutions for combinatorial problems but it is very difficult to find provably optimal ones.

Linear programming is a type of mathematical programming where the objective function and the variable constraint functions are all linear and the decision variables may take on continuous non-negative real values. Note that linear programming relates to the process of solving a linear program, which is an instance of a linear programming problem. Both linear programs and programming are abbreviated using LP. Integer linear programming (ILP) and mixed integer programming (MIP) are subtypes of linear programming where the decision variables may only take on integer values or where some, but not all, decision variables may take on integer values respectively. An additional designation that may be added to either integer or mixed integer programs is the "1/0" prefix. Adding this prefix when referring to an ILP or a MIP problem means that at least one of the integer variables in the problem may only take on binary  $\{0, 1\}$  values. Variables constrained in this way typically represent no/yes decision outcomes. In this thesis, the distinction between ILP and MIP is not very

important as they are treated equally when the problems are solved. For this reason, for the rest of the thesis, both types will be referred to as ILPs.

### 2.12.3 Formulation of an LP/ILP model

Each LP/ILP formulation has the following main components: the data set, the variables of the problem, the constraint and variable bound functions and lastly the objective function. These individual components will be explained using a classical combinatorial optimization problem called the 1/0 knapsack problem [48]. The knapsack problem is actually an ILP problem, but we will use it as an example for both LPs and ILPs as their formulations are nearly identical. The only difference between the two formulations is that the variables are constrained to be integers in one and not the other.

Consider a problem of a person who must pack for a hike. Assume that she has several unique packages of varying utility and weight. The problem is to determine which items the hiker should pack so that the carrying capacity for the hiker is not exceeded while the total utility of all the items packed is maximized. The data set in this problem comprises of the individual items and their properties, such as the pack's size and the weights and utility values for the items. The data set can be split into two separate parts: sets and parameters. A set is a collection of elements that is to be considered in the optimization and parameters are input data that is known in advance. The parameters may be constants (such as the maximum weight that can be carried by the hiker) or properties of the objects in the sets (such as weight or utility). In this example, the data set is defined as follows:

#### **Sets:**

*I*                    The set of items that the hiker may pack in her bag.

#### **Parameters:**

$u^i$  This parameter corresponds to the utility of item  $i$ .  $i \in I$ .

$w^i$  This parameter corresponds to the weight of item  $i$ .  $i \in I$ .

$W$  This parameter corresponds to the carrying capacity of the hiker.

The variables are the unknowns of the problem. In this example, the variables will be yes/no decisions that have to be made by the hiker for each item, defined as follows:

**Variables:**

$x^i$  Equal to 1 if item  $i$  is to be packed, and is equal to 0 otherwise.

The objective function is a function of the decision variables that represents the main goal of the problem. In this case, it represents the total utility of all the items included in the pack, which is what this problem attempts to maximize is given by (2.3).

**Objective Function:**

$$\text{Maximize: } \sum_{i \in I} u^i \cdot x^i \tag{2.3}$$

The constraint and bound functions assert relationships between variables that must be adhered to. In the case of this example, it is important that we impose a limit on the weight carried by the hiker or the solver, in an attempt to maximize the objective function, will select every single item to be packed which is undesirable. The constraint functions asserting that the maximum carrying capacity of the hiker may not

be exceeded is given by (2.4). In order to make sure that the solver only selects values of 1 or 0 for the decision variables, the variable bound function is given by (2.5).

**Constraints:**

$$\sum_{i \in I} w^i \cdot x^i \leq W \tag{2.4}$$

$$x^i \in \{0, 1\} \tag{2.5}$$

For example, consider a problem above in which the hiker has an inventory of four items where  $w^i = \{10, 30, 15, 20\}$ , corresponding to the weight of each item in kilograms, and  $u^i = \{12, 10, 9, 6\}$ , corresponding to the number of days the hiker may travel if she packs item  $i$ . Suppose that she wishes that the weight of the pack did not exceed 45 kg. If she attempts to pack her backpack by picking the items with the most utility first, she will end up with a solution where  $x^i = \{1, 1, 0, 0\}$ , which will allow her to travel for 22 days carrying a pack weighing 40 kg. The optimal solution to this problem, which can be obtained using the ILP methods introduced below, is  $x^i = \{1, 0, 1, 1\}$ , corresponding to a travel time of 27 days and a pack weighing 45 kg. Note that in this case, due to the small problem size, it is fairly easy to find the optimal solution simply by enumerating all possible packing arrangements. This would not be practical for larger sizes of the problem, considering that the number of possible item arrangements grows exponentially (in the order of  $2^n$ ) as more variables  $n$  are added.

**2.12.4 Solving LP and ILP Problems**

LP problems are most often solved to optimality using a simple and efficient method called the simplex method [49]. The simplex method, examined in closer

detail below, involves a series of sequential matrix operations on a representation of the LP problem in standard matrix form. ILP problems (such as the knapsack problem, above) are typically more difficult to solve than LPs and cannot be directly solved using the simplex method. The reason for this is because it is not possible to represent discrete integer values in the simplex method. Typically, solving an ILP starts by solving an LP relaxation of the ILP problem, which is a formulation of the ILP where the integrality constraints on variables are relaxed. The solution to the relaxation provides an upper or lower bound on the objective value of the ILP, based on whether it is a maximization or a minimization problem, respectively. If the solution obtained at this stage is an integer solution then the process terminates and the solution can be said to be an optimal integer solution.

If not, the most common approach is to use the branch-and-bound (B&B) method, originally proposed in 1960 [50]. This method works by solving a sequence of LPs obtained by relaxing integrality constraints of the problem and adding new constraints to separate the feasible region into sub-regions to be used in subsequent iterations of the algorithm. Typically the implementation of the B&B method is viewed as a tree search and the number of new constraints added increases as the algorithm progresses. Other approaches for solving ILPs such as the Gomory cuts/cutting plane method [51] or the hybrid branch and cut method [52] use essentially the same idea: that of iteratively adding constraints to the LP relaxation of the problem and resolving it until all the required variables are integer. They mainly differ by the types of constraints, also referred to as cuts, that are used at each iteration.

Regardless of the method used, it proceeds until all the necessary variables are integer, or until a specific exit criteria, such as reaching a solution within the MIPGAP, is met. The MIPGAP is the allowed difference between the objective value of the fully relaxed LP lower (minimization) or upper (maximization) bound solution and the best currently found integer solution. This difference can be defined in absolute units or as a percentage. Thus, for example, if an ILP problem with the MIPGAP

set at 1% terminates, the integer solution found is provably within 1% of the true optimal solution. Due to their combinatorial nature, the solution space of the ILP problems addressed in this thesis are very large and may contain many near-optimal solutions. Searching this space to prove that the current integer solution is indeed the optimal one may take a very long time as every possible alternative may have to be examined. In network design engineering problems, it is perfectly acceptable for a solution to be within 1% of the optimal. In fact, for network design problems, even solutions obtained that are within 50% MIPGAP are generally better than ones obtained using heuristic methods and can be obtained much faster, considering the time needed to develop and implement a good quality heuristic method.

It is possible to solve LP and ILP problems by hand using the methods mentioned above for problems with just a few variables and constraints. This approach would not be practical for the size of the problems in this thesis which contain tens of thousands of variables and constraints. Because of this, highly specialized and efficient third party packages that implement the above methods are used to generate optimal solutions. AMPL [53] (an acronym for ‘a mathematical programming language’) is a commercial software package and modeling language that can be used to describe mathematical programs in the format outlined in Section 2.12.3, above. The AMPL software, using the problem formulation as well as the required data sets, prepares the problem so that it can be passed to CPLEX. CPLEX [54] (named after the simplex method and the C programming language) is a sophisticated commercial software package designed to solve mathematical programming problems. It is then used to find the optimal or near optimal solution to the defined problem.

### **2.12.5 The Simplex Method**

The simplex method (developed in 1947 by George Danzig) [49] is a very common and efficient iterative method of solving LP problems. The aim of this section is to acquaint the reader with how the method works as well as relevant terminol-

ogy and to aid in the understanding of the column generation method, introduced below. This section will provide a small example to illustrate how the simplex method works. For detailed discussion, including proofs and variations of the simplex method, the reader is deferred to [45], [55].

Consider the LP problem below, in standard form, where the objective function is given by (2.6) and the constraints of the problem are given by (2.7). The bounds (2.8) ensure that the variables do not take on negative values.

$$\text{Minimize: } z = 5x_1 + 3x_2 \tag{2.6}$$

$$\begin{aligned} \text{Subject to: } & 2x_1 + 1x_2 \leq 10 \\ & 3x_1 + 2x_2 \leq 16 \end{aligned} \tag{2.7}$$

$$x_1, x_2 \geq 0 \tag{2.8}$$

The simplex method cannot be applied directly to the LP above because it contains inequality constraints. A preliminary step of the method consists of introducing so-called slack variables (called surplus variables for minimization problems) to the constraints. The slack variables are added one per inequality constraint and represent the difference between the left and right hand sides of the constraint equations. These slack variables are also added to the objective function, but their coefficients are set to 0, representing that they do not affect the objective value. Equations (2.9), (2.10) and (2.11) illustrate the resulting formulation of the LP, in what is referred to as augmented form.

$$\text{Minimize: } z = 5x_1 + 3x_2 + 0s_1 + 0s_2 \tag{2.9}$$

$$\begin{aligned} \text{Subject to: } & 2x_1 + 1x_2 + 1s_1 + 0s_2 = 10 \\ & 3x_1 + 2x_2 + 0s_1 + 1s_2 = 16 \end{aligned} \tag{2.10}$$

$$x_1, x_2, s_1, s_2 \geq 0 \tag{2.11}$$

As a final step in initiating the simplex method, it is common to place the coefficients of the LP problem in the augmented form into a tableau. The resulting tableau is shown in (2.12) where the first two constraints are represented by the first two rows of the matrix and the objective function (with the sign reversed) is represented by the last row.

$$\begin{array}{ccccc}
 x_1 & x_2 & s_1 & s_2 & RHS \\
 2 & 1 & 1 & 0 & 10 \\
 3 & 2 & 0 & 1 & 16 \\
 -5 & -3 & 0 & 0 & 0
 \end{array} \tag{2.12}$$

Each column in the tableau above, with the exception of the last one, corresponds to the specified variable. The last column, labeled as the right hand side (RHS), represents the values of the variables currently in the basis as well as the current objective value,  $(-z)$ , which is found in the bottom most element of the RHS column. The basis represents a set of variables (called basic variables) that make up a solution for the LP problem. Variables that are not in the basis are referred to as non-basic variables and their values are set to 0. The basis may correspond to a feasible or an infeasible solution, depending on whether the values of the basic variables adhere to the constraints of the LP or not, respectively. It may also correspond to an unbounded solution if any of the variables affecting the optimal value are unconstrained. If any of the basic variables have a value of 0, then the corresponding basic solution is said to be degenerate. For our purposes of illustration, we will only deal with feasible basic non-degenerate solutions in this example.

In (2.12), the basic variables are  $s_1$  and  $s_2$  and their values are 10 and 16, respectively. The basic variables can be identified by examining the tableau for columns where a single element is set to 1 and the rest are set to 0. The row containing the non-zero element can be used to determine the value of the corresponding basic variable, simply by looking in the RHS column of that row. The non-basic variables

are  $x_1$  and  $x_2$  and their values are 0, as is the case for any variable not in the basis.

The objective value corresponding to this solution is  $-z = 0$ .

The last row of the tableau has a very important role in the context of the simplex method. The elements of this row, excluding the element in the RHS column, correspond to the sensitivity of the objective value to a unit increase in this element's column variable. These elements are referred to as reduced costs of the variable in question.

The grand strategy of the simplex method is that of successive improvements. Having found an initial basic feasible solution, the simplex method searches for other feasible solutions with the main goal being the improvement of the objective value. This is done by, first identifying if there are any non-basic variables that can improve the objective value and picking the variable that is most efficient at doing so to enter the basis. The column corresponding to this variable is referred to as the pivot column. Effectively, the value of this variable is increased, while keeping the values of the other non-basic variables at 0, until some non-negativity constraint of one of the currently basic variables restricts the entering variable from being increased further. The variable that imposes this bound on the entering variable is picked to exit the basis. The next step of the simplex method entails modifying the tableau to account for the new basic and non-basic variables. This is done by applying a series of elementary operations to the tableau such that the columns corresponding to the basic variables contain a single element set at 1 and the rest of the elements in the column set at 0. This process is repeated until no more variables can be found to enter the basis. At this point, the current solution corresponds to the optimal one and the process terminates.

Applying the above process to our example, we first need to select a variable to enter the basis. For this we can select  $x_1$  since it has the highest negative reduced cost of -5 units so increasing  $x_1$  will have the biggest effect on the objective value. The

variable providing the most stringent upper bound on  $x_1$  is  $s_1$ . This is determined by identifying which row has the smallest positive ratio  $\left(\frac{s}{r}\right)$  where  $s$  is the value of the current basic variable as read off the RHS column and  $r$  is the value of the row coefficient of the column corresponding to the variable entering the basis. In our example  $\frac{10}{2} < \frac{16}{3}$  meaning that the first row must be selected as the so-called pivot row, corresponding to  $s_1$  leaving the basis. The updated tableau, after elementary row operations are used account for the changes in the basis, is shown in (2.13).

$$\begin{array}{cccccc}
 x_1 & x_2 & s_1 & s_2 & RHS & \\
 1 & \frac{1}{2} & \frac{1}{2} & 0 & 5 & \\
 0 & \frac{1}{2} & -\frac{3}{2} & 1 & 1 & \\
 0 & -\frac{1}{2} & \frac{5}{2} & 0 & -25 & 
 \end{array} \tag{2.13}$$

Note that now  $x_1$  and  $s_2$  are basic variables with values of 5 and 1 respectively. The objective value of this solution is  $-z = 25$ , but this is not yet optimal, judging by the fact that there remains one more column (corresponding to  $x_2$ ) where the reduced cost can still contribute to improving the objective value. With  $x_2$  entering the basis, we need to select a variable to exit the basis. The smallest  $\left(\frac{s}{r}\right)$  ratio belongs to the second row, meaning that  $s_2$  must leave the basis. The updated final tableau after the pivot step is shown in (2.14).

$$\begin{array}{ccccc}
x_1 & x_2 & s_1 & s_2 & RHS \\
1 & 0 & 2 & -1 & 4 \\
0 & 1 & -3 & 2 & 2 \\
0 & 0 & 1 & 1 & -26
\end{array} \tag{2.14}$$

In the final tableau,  $x_1$  and  $x_2$  are the basic variables, and their values are 4 and 2, respectively. There are no more columns with a reduced cost that can improve the objective value, meaning that the solution is indeed optimal and the objective value is  $-z = 26$ .

The above was a very simple green field example of the simplex method. In practice, there may be several hurdles that may need to be overcome in order to arrive at a solution. For example, it may not be a trivial task to arrive at a feasible solution to be used as a starting point for the simplex algorithm. Also, during execution the algorithm may not be able to find a variable to enter the basis or one or more basic variables may have values of 0, corresponding to a solution that is degenerate. Both of the above may lead to cycling and prevent the simplex method from finding an optimal solution or terminating.

Additionally, it is worth mentioning that there exists another very popular version of the simplex method called the revised simplex method. Essentially, it acknowledges that at any iteration of the simplex method the major players are the variables entering and exiting the basis, as well as the basis itself. In the revised simplex method, it is no longer necessary to perform the intensely computational task of updating all the columns in the tableau at the end of every iteration. Instead the algorithm maintains a matrix containing the columns of the basic variables, the inverse to which must be found on every iteration in order for the algorithm to proceed forward. Despite this, typically an iteration of the revised simplex takes less time than that of the simplex method and for this reason, the revised simplex method is more commonly found in computer programs for solving LP problems. Further discussion of the revised sim-

plex methods as well as methods for identifying and overcoming the pitfalls mentioned above are out of the scope of this thesis. The reader is referred to [45], [55] for further discussion on these topics.

### 2.12.6 Duality

An interesting property of linear programming is that every maximization problem in standard form (referred to as the primal) gives rise to a minimization LP problem called the dual problem. Consider a primal LP problem with  $n$  variables and  $m$  constraints in standard form presented using (2.15), (2.16) and (2.17). The dual of this LP problem, presented using (2.18), (2.19) and (2.20) has  $m$  variables (referred to as dual variables  $y_i$ ) and  $n$  constraints. Note that the direction of optimization is reversed, the objective function variable coefficients  $c_j$  and the right hand side constraint values  $b_i$  of the primal switch positions in the dual. The constraint coefficient matrix is effectively transposed in the dual problem. The two LP problems are, in fact, duals of one another and are related in an interesting way: a feasible solution in one yields a bound on the optimal value in the other. Furthermore their objective values coincide at optimality.

$$\text{Maximize: } \sum_{j=1}^n c_j \cdot x_j \quad (2.15)$$

$$\text{Subject to: } \sum_{j=1}^n a_{i,j} \cdot x_j \leq b_i \quad (i = 1, 2, \dots, m) \quad (2.16)$$

$$x_j \geq 0 \quad (j = 1, 2, \dots, n) \quad (2.17)$$

$$\text{Minimize: } \sum_{i=1}^m b_i \cdot y_i \quad (2.18)$$

$$\text{Subject to: } \sum_{j=1}^n a_{i,j} \cdot y_j \geq c_i \quad (i = 1, 2, \dots, m) \quad (2.19)$$

$$y_j \geq 0 \quad (j = 1, 2, \dots, n) \quad (2.20)$$

The property of duality in LP problems has some important uses. It can allow us to check whether a solution arrived at by solving the primal problem is indeed optimal. If it isn't, the corresponding dual solution will be infeasible. This property works in the reverse direction as well: a sub-optimal solution in the dual corresponds to an infeasible solution in the primal. Visually, the two problems can be seen as approaching the same optimal solution from different directions. The primal is concerned with searching the feasible solution space (which takes the shape of a convex polytope), moving from one feasible solution to the next, while improving the objective value. The dual approaches the same solution from a region of infeasibility (from the perspective of the primal) and the two coincide at optimality. What is particularly interesting is that the dual solution can be extracted from the final tableau of the primal problem, which is a consequence of what is called the complimentary slackness theorem. This means that the benefits that come from knowing the dual solution can be gained without even solving the dual problem.

Duality also allows a problem that would be difficult to solve in the primal to be easily solved in the dual under certain criteria. An example of this is a situation where the primal problem has many more constraints than variables. Using the simplex method, many rows (each corresponding to a constraint) would have to be updated at every iteration making solving the primal computationally intensive. In the corresponding dual problem, there would be many more variables than constraints, which is a favourable situation for the simplex method, meaning that the dual problem can be solved much faster than the primal and would yield the same optimal solution.

Another interesting property of duality is found in the meaning of the dual variables. For example consider the primal problem (in standard form (2.15), (2.16) and (2.17)) of maximizing the profit of a furniture firm. Suppose the primal variables  $x_j$  represent the quantity of an item of furniture to be built (chairs, tables) and the coefficients  $c_j$  represent the profit that can be made per item. Each constraint  $i$  represents a resource such as wood or capital. The value of  $b_i$  corresponds to the total amount of resource  $i$  available and the coefficients  $a_{i,j}$  represent the amount of resource  $i$  consumed in the production of making an item  $j$ . Consider that the dual of the problem is given by (2.18), (2.19) and (2.20). The units of the primal objective function (2.15) are in dollars, and the same can be said about the dual objective function (2.18) since they seek the same optimal value. We know that coefficients  $b_i$  in (2.18) are in units of resource  $i$ . For the units to work out correctly,  $y_i$  must be in units of dollars per resource. The same logic applies if the units on the left hand side and the right hand side of (2.19) are to match. Effectively, variables  $y_i$  measure the unit worth of each resource. By examining  $y_i$  we can determine how the objective value would be affected by relaxing the constraint represented by the dual variable by one unit. For example, suppose that the furniture company consumes a certain amount of wood per week. And suppose that the value of the dual variable (also referred to as a shadow price) corresponding to the value per unit of wood is \$10. It can be inferred that should the management decide to order more wood, they should make sure not to pay more than \$10 per unit or the action will not be a profitable one. It should be noted that this kind of analysis is most accurate for small changes in the resource, and that its conclusions may fail when the changes are large. Despite this, this type of analysis is highly valuable in business applications.

### 2.12.7 Column Generation

One particular concept related to the simplex method, relevant to Chapter 4, is the concept of column generation. The size of the LP problem that can be solved using the simplex method is ultimately limited by the computational resources available, such as computer memory. The column generation approach offers a method for solving LPs with a large number of variables and a relatively small constraint set without having to generate and update a large number of columns or having to find an inverse to a very large matrix (a step required in the revised simplex method). It provides a way for the LP to be broken down into two sub-problems: the master problem and the pricing problem. The master problem is the original formulation with only a subset of the variables considered. The pricing problem is a new problem created to generate new variables (corresponding to columns) with the aim of identifying the most useful columns to be added to master problem.

In the simplex method, the criteria for a column to enter the basis is that it must have a negative reduced cost. It is possible to calculate the reduced cost  $r_j$  of a non-basic column  $j$  of the tableau form of the LP using equation (2.21). In this equation  $c_j$  is the coefficient corresponding to the column  $j$  in the objective function,  $c_B^T$  is the row matrix containing the objective function coefficients of the basic variables,  $B^{-1}$  is the inverse of the constraint coefficient matrix containing the columns of the basic variables and  $A_j$  is the constraint coefficient column  $j$ .

$$r_j = c_j - c_B^T \cdot B^{-1} \cdot A_j \quad (2.21)$$

In the simplex method, this reduced cost is readily available and can be calculated for any non-basic column easily. In the tableau form the reduced costs were implicitly calculated by the row operations used to update the matrix of the LP in Section 2.12.5. As was mentioned before, it is possible to obtain the dual variables from the final tableau of the primal problem. This can be done by using

$y^T = c_B^T \cdot B^{-1}$  where  $y^T$  is the vector corresponding to the dual variables. Combining this with (2.21) we get:

$$r_j = c_j - y^T \cdot A_j \quad (2.22)$$

The way that column generation works is that the master problem is first solved and the dual variables corresponding to it are calculated. The dual variables are passed to the pricing problem to solve for the column  $A_j$  while minimizing the reduced cost, given by (2.22). If a solution to the pricing problem with a negative reduced cost exists, then this column is added to the master problem where it can enter the basis. The master problem is re-solved, new dual variables are generated and the process proceeds to the next iteration. If the pricing problem cannot find any solutions with a negative reduced cost, then the solution found by the primal problem is said to be optimal.

The column generation approach can be understood in an intuitive way if the problem can be formulated in a way where a single column represents a pattern or a combination of smaller parts and the goal of the master problem is to find a combination of the patterns such that some criteria is met. A very good example of this is the cutting-stock problem. In this problem the goal is to cut a number of same length raws into a set of pieces of pre-determined sizes such that waste produced by the process is minimized. In this problem it is possible to express each of the different ways of cutting a single raw as a column in the LP. The goal of master problem is to try to find which of these cutting patterns should be used and in what quantity. When dual variables are extracted from the master problem, they effectively represent the value certain resources have to the master problem. The pricing problem will take these into consideration and come up with a solution that utilizes this information while also adhering to the constraints imposed by the original problem (that the length of the raw may not be exceeded, for example). Effectively, the pricing problem to the cutting

stock problem, takes on the form of the knapsack problem from Section 2.12.4, where the dual variables correspond to the item utilities and cut lengths correspond to item weights.

## 2.13 Algorithms

The ILP formulations in this thesis often require that a data set containing working routing as well as a set of eligible cycles is provided to the solver in addition to the ILP model. Doing so relieves the solver of having to generate this data as part of the optimization process, typically greatly reducing the complexity of the problem. This section outlines the main graph search algorithms used to generate these data sets.

### 2.13.1 Dijkstra's Algorithm

Dijkstra's algorithm [56] is commonly used to find the shortest (lowest cost) path between a select vertex (source node) and all other vertices (nodes) in a network graph. In its original form, it is best suitable for network problems where the edges (spans) are assigned non-negative weights (costs, lengths). It works by giving each node a label that is updated during the run of the algorithm and can be either permanent or temporary. The labels contain two pieces of information: the total cost of the path from the source node and the name of the previous node crossed by the path from the source node to the label's node. These labels, which are blank initially, are updated during the run time of the algorithm. The source node always contains the distance 0 in its label. The algorithm works by picking a *current* node (which is initially the source node) and making its label permanent. It then examines all the adjacent nodes with temporary labels and updates them if they are blank or if the total distance recorded by the temporary label is greater than the path taken to it from the source node through the *current* node. When all the adjacent nodes with temporary labels have been examined (and updated if need be), the algorithm picks the node

with the label containing the shortest distance as the next *current* node. The algorithm repeats until there are no more nodes with temporary labels. When the algorithm completes, every node's label contains the total cost of the shortest path to the source node as well as the previous node crossed by the path. Starting at any node, it is possible to immediately know the cost of the path from that node to the source node and by following the node labels back to the source node, the actual path taken can be extracted.

A very basic example of how Dijkstra's algorithm works is shown in Figure 2.15. In this example, node A is selected to be the source node. Initially, its label is set to [0,-] and is made permanent (labeled red) while all the other node labels remain temporary (labeled blue). At the end of the first iteration, as illustrated by Figure 2.15a, the labels for node B and C have been updated and now contain the distance of the shortest path en route to them from the source node that has been found so far. Their labels also contain the previous node that path crossed, which is A for both nodes. The shortest of the two paths leads to B, so for the next iteration it becomes the *current* node and its label is made permanent. The second iteration ends after all the temporary nodes adjacent to B have been examined as illustrated by Figure 2.15b. The obvious choice for the next *current* node is C. Its label is made permanent and all of the adjacent nodes to it with temporary labels are examined. Note that in this iteration, node D's label was updated from [6,B] to [5,C] indicating that a new shorter path was found when C became the *current* node. Since after node D's label is set to permanent there are no more nodes with temporary labels, the algorithm terminates.

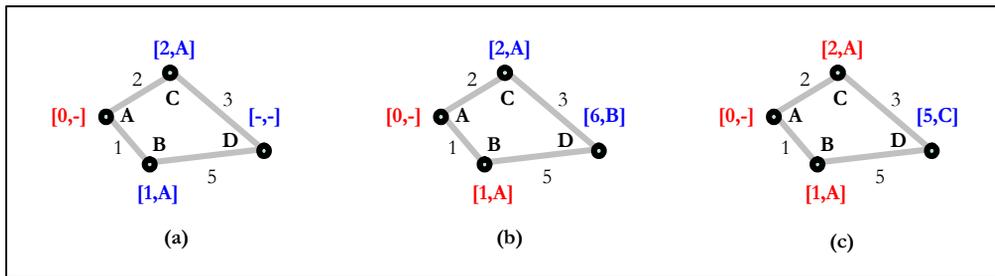


FIGURE 2.15. Dijkstra's algorithm Example.

The basic version of Dijkstra's algorithm only works on network graphs with non-negative edges. Given negatively weighted edges, it fails to find the shortest path because nodes end up being labeled as permanent before the shortest path to them can be found. The variation on Dijkstra's algorithm proposed in [57] makes sure that all nodes adjacent to the current node are scanned, not just the ones with the temporary label. Any node labeled permanent that gets updated in this step gets labeled temporary so that it may be considered to be the next current node and the nodes adjacent to it get an opportunity to be updated as well. It is also possible to speed up Dijkstra's algorithm as proposed in [57] by using the breadth-first search (BFS) Dijkstra's algorithm. It is best suited to finding the shortest path between the source node and a single destination node. It works by scanning from every node that was updated in the previous iteration. It effectively does in parallel what regular Dijkstra's algorithm does sequentially. As soon as the destination node receives a label, every other node whose label has a higher value than the label at the destination node is discarded. This means that the size of the graph is dramatically reduced as soon as the destination node is first reached and the shortest path to the destination node is found shortly after as the algorithm terminates.

### 2.13.2 Cycle Generation - Depth First Search

Many methods for  $p$ -cycle design require that the set of cycles eligible for use in the design is provided as input to the problem. This section describes a cycle finding algorithm that uses the depth first search approach and is able to find every cycle in a given network graph [58], [59]. The algorithm works by first selecting a root node  $x$  and then selecting an adjacent node  $y_0$  and marking it so that it cannot be selected again by the algorithm. An unmarked node adjacent to  $y_0$  is selected and marked  $y_1$  and so on to node  $y_k$ . This process stops when one of three things happen: a) the root node  $x$  is reached ( $y_{k+1} = x$ ), b) all of the nodes adjacent to node  $y_k$  are marked, or c) the specified limit on cycle size is exceeded. Encountering the root node  $x$  means that a cycle has been found. Once any of the three conditions happen, the algorithm retracts to the previous node and proceeds to the next unlabeled node if one is available to search for more cycles. In the process of backing up, the node being backed up from remains marked in the case b) so that further searching is prohibited in this direction. However, in cases a) and c) the node being backed up from is unmarked (along with all the nodes that node blocked by being marked) so that searching may continue and alternate cycles that are within the specified cycle size can be found. This whole process continues until the algorithm backs up all the way back to the root node  $x$ . At this point, every cycle crossing the span  $(x, y_0)$  has been found. This span is removed from the network graph and a new node adjacent to  $x$  is selected and labeled as  $y_0$  and the depth first search starts again. When only one span adjacent to  $x$  remains, it and the root node are removed from the network graph and a new root node is selected. The whole process terminates when there is only one node left in the network graph.

## 2.14 Summary

In this chapter we introduced basic concepts of transport networking. We also introduced the concept of optical networking as well as relevant equipment and technologies. The topic of network survivability was extensively covered, including a thorough summary of the survivability techniques most covered in literature. In this chapter concepts of optimization, graph and set theory as well as routing algorithms relevant to this thesis were presented. The following chapter introduces the concept of network survivability using FIPP  $p$ -cycles which is the main theme of this thesis.

## 3.1 Introduction

The following chapter introduces the FIPP  $p$ -cycle concept, which is the main focus of this thesis. Here FIPP  $p$ -cycles are introduced in thorough detail with specific attention paid to the concept of pre-connection and its importance in optical transport networking. The different types of cycle to working path relationships are introduced. This is followed by a breakdown of the configuration types possible under the FIPP  $p$ -cycle architecture. The chapter concludes with a thorough literature review of the subject.

## 3.2 Background

Failure-independent path-protecting (FIPP)  $p$ -cycles is a recently introduced network architecture that extends the concept of span-protecting  $p$ -cycles to the protection of end-to-end paths. Span-protecting  $p$ -cycles is a well known network architecture that has properties desirable in transport networking such as high capacity efficiency and fast restoration times. The fast restoration times are due to the fact that, similar to ring systems,  $p$ -cycles are cyclical and pre-connected structures, where the restoration action as well as the replacement paths provided by each cycle are known prior to failure. Because of this, the actual switching actions are completely pre-planned and are no more complex than that of ring systems. The high efficiency of  $p$ -cycles comes from their ability to protect straddling spans. These are spans that are not crossed by the cycle, but on which the cycle is able to provide protection for two channels of working capacity. The quote “ring-like speed with mesh-like capacity” [35] is often used to describe  $p$ -cycles in literature and the main reason as to why this architecture is so widely and extensively researched.

FIPP  $p$ -cycles retain the fast restoration times and high capacity efficiency of  $p$ -cycles but protect entire end-to-end paths instead of individual spans. Where span-protecting  $p$ -cycles provided replacement paths for channels on failed spans, FIPP  $p$ -cycles provide entire end-to-end protection paths for working paths. Only the end-nodes of the working path are responsible for failure detection and the switching actions required for failure recovery. This makes FIPP  $p$ -cycles as fast as span-protecting  $p$ -cycles, and much faster than a pre-planned, but not pre-connected path-protecting architecture such as SBPP. The location of the actual failure is not important in order for the restoration action to take place; only the knowledge that the failure has occurred is required. Straddler protection is also inherited by FIPP  $p$ -cycles, but where span-protecting  $p$ -cycles protected two channels on a straddling span, it is possible to protect up to two working paths on a route straddling a FIPP  $p$ -cycle. This contributes to the high capacity efficiency of this architecture as it allows a large number of working paths to share a single FIPP  $p$ -cycle for protection.

One very important property of the FIPP  $p$ -cycle architecture is the fact that it allows for entire protection paths to be pre-cross-connected. This property of pre-connection of protection paths is uniquely important to and especially desirable in transparent optical networks [33], [39] where pre-connection means that the protection paths may be pre-engineered, tested and in an operational, stable state prior to failure. This is important because using modern commercial optical and switching network technology it is not realistic to expect that an arbitrary set of optical or wavelength channels on a multi-hop protection path can be concatenated in real-time and with acceptably low bit error rate (BER). This is considering that there are dozens of impairments such as polarization, dispersion, amplifier gain transients, intermodulation, power levels, noise and nonlinearities that must be carefully engineered before an optical path of sufficient transmission quality in a DWDM environment can be established [60].

### 3.3 Working Path/Cycle Relationships

A typical FIPP  $p$ -cycle network design contains several cycles, with each cycle protecting a set of working paths. Each cycle and a working path set protected by this cycle is referred to as a configuration. In literature, the term configuration can also mean a cycle and a route set (as opposed to path set). In general, the two can be used interchangeably, because when the term route set is used, a path set is implied. In the context of FIPP  $p$ -cycles, the route set naming convention is short hand used to indicate a specific type of path set. The differences between the two are easier explained in the context of different basic cycle/working route relationships possible in FIPP  $p$ -cycles, which are introduced below.

#### 3.3.1 On-Cycle Relationship

An “on-cycle” relationship is said to occur between a working path and a FIPP  $p$ -cycle when the path and the  $p$ -cycle assigned to protect it have one or more spans in common. When a working path shares all of its spans with the cycle, then it is said to be in a fully on-cycle relationship. A partially on-cycle relationship occurs if some of the spans on the working path are not shared by the cycle. An on-cycle working path can be protected by a cycle passing through the end-nodes of that path. Operationally, a FIPP  $p$ -cycle treats both types of on-cycle relationship the same; the backup path is provided on the side of the cycle that does not cross any of the same spans as the working path. An on-cycle working path can also be referred to as an on-cycle working route and the two terms are interchangeable. To illustrate, Figure 3.1a shows a partially on-cycle working path being protected by a FIPP  $p$ -cycle, and Figure 3.1b shows the protection path after the failure where the dashed arrowed curve represents the failed working path and the light arrowed curve represents the enabled restoration path.

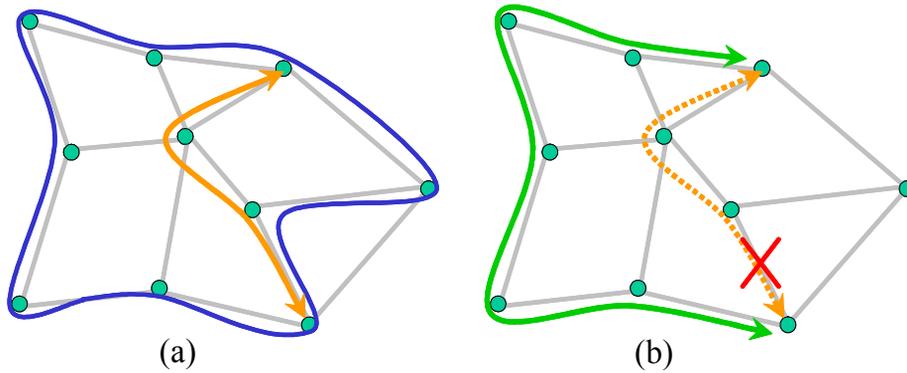
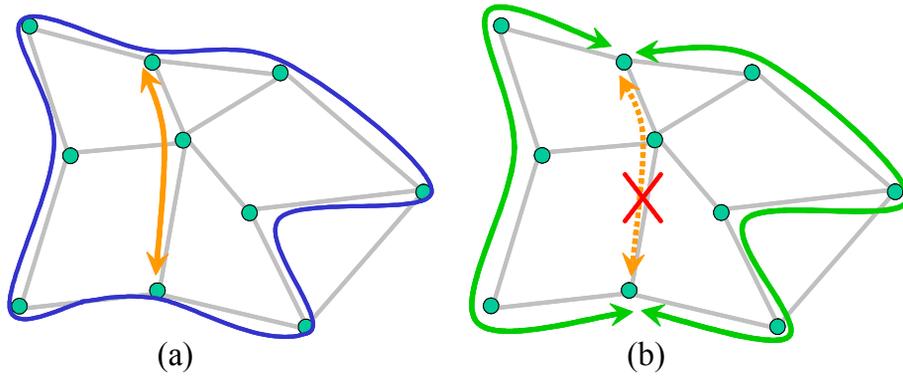


FIGURE 3.1. a) On-cycle protection relationship and b) the restoration action taken after the working path fails.

### 3.3.2 Straddling Relationship

A working path is said to be in a straddling relationship with a cycle when the two are mutually span disjoint and where the cycle passes through the end-nodes of the path. A unit straddling working path protected by a cycle has two eligible protection paths: the end-nodes of the working path effectively split the cycle into two path segments that can be used to restore the working path if it fails. Two identical straddling paths can be protected by the same cycle simultaneously using the two path segments provided by the cycle. In FIPP literature, these two paths as a unit are commonly referred to as a straddling route. It can be said that the cycle is able to provide two protection paths to any straddling route and thus is able to protect up to two working paths on it, per unit of  $p$ -cycle capacity. If the working route in this relationship fails, then its end-nodes simply break into both sides of the cycle and thus divert the traffic along intact protection paths on that cycle. This relationship is illustrated in Figure 3.2a which shows a single straddling route protected by a unit cycle. The two protection paths that are provided by the cycle and enabled after a failure are shown in Figure 3.2b.



**FIGURE 3.2. a) Straddling protection relationship and b) restoration paths for a straddling route.**

Note that the term ‘straddling route’ is intended to be used in the same way that the term ‘straddling span’ is used in the context of span-protecting  $p$ -cycles. Straddling span does not imply that the cycle protects the span in its entirety, but only that at most two working channels are protected by the cycle on this span. Straddling route is a lot easier to say than ‘two identically routed paths protected by the same cycle’ and this nomenclature is used throughout this thesis.

### 3.3.3 z-Case

The z-case, illustrated in Figure 3.3, is a special case of the partially on-cycle relationship. Here the working path and the cycle cross the same spans in such a way that both of the potential protection paths provided by the cycle are non-disjoint from the working path. This means that any time a working path fails, one of the two eligible protection paths provided by the cycle may fail as well. Essentially, where a working path is protected by a cycle in a z-case relationship, it must have access to both possible protection paths provided by the cycle, and exactly which path will be taken is only known after a failure occurs. The assumption in literature is that the nodal equipment is capable of handling this situation. Of the two eligible protection paths, one is selected as the primary and is switched to any time it and the working path it

protects do not fail simultaneously. In the case that they do, the secondary protection path is enabled instead. The mechanism for detecting whether the primary protection path has failed or not is exactly the same as the mechanism used to detect a working path failure. In transparent optical networks, the end-nodes can react to whenever a loss of light occurs on a particular channel be it working or protection, indicating path failure. The only time *z*-cases become problematic is under a constraint where every working path is required to have a single protection path that is switched to regardless of where the working path is affected by failure: a condition that *z*-cases violate. Ultimately this means that if the hardware cannot support the *z*-case relationship, any design that allows these relationships to exist will not be 100% single failure restorable.

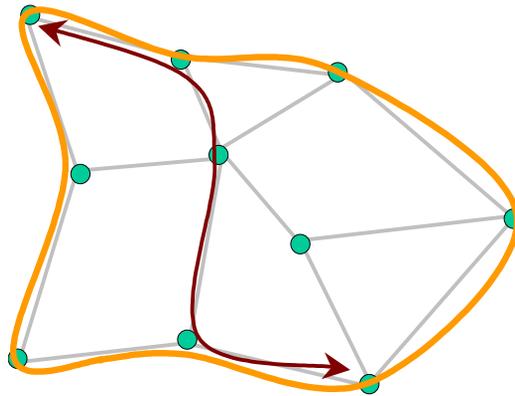


FIGURE 3.3. *z*-Case. FIPP *p*-cycle and working path are shown as a connected thick line and an arrowed line, respectively.

### 3.4 Configurations

As mentioned before, each FIPP configuration contains a unit cycle as well as a set of working paths whose protection paths are provided using the spare capacity reserved for the cycle. In general, for a working path to be eligible for protection by a cycle, the end-nodes of the working path must be crossed by the cycle. For a FIPP design to be 100% single span failure restorable, A: the working paths protected by a

same cycle must be mutually span disjoint from one another or B: if the working paths are not disjoint, their protection paths provided by the cycle must be. Working paths that are mutually span disjoint cannot be affected by the same single span failure and thus cannot compete for spare capacity provided by the  $p$ -cycle. However, if two working paths are not mutually span disjoint and can fail simultaneously (such as in the case of straddling routes), their protection paths must be mutually span disjoint because a unit cycle cannot simultaneously support two enabled protection paths crossing the same span. It is worth noting that, in literature, configurations that contain configurations that adhere to criteria A only are referred to as Type-1 configurations while those that allow for both A and B are referred to as Type-2.

The above criteria is general and defines exactly the relationships that working paths must have to one another and the cycle protecting them. It applies to path protection using  $p$ -cycles in general, of which FIPP is actually a sub-class. As will be seen later in this thesis, this criteria is too general in terms of practical implementation of FIPP  $p$ -cycles. Because of this, (and the need for practical methods for FIPP  $p$ -cycle design) the concept of a disjoint route set (DRS) was introduced. This concept is based on the observation that a set of mutually span disjoint routes can share a  $p$ -cycle as long as the cycle crosses all the end-nodes of the working routes in the set. The type of protection relationship provided for each route by the cycle (on-cycle or straddler) is based entirely on whether the cycle is span disjoint with the route or not. The specific consequences and benefits that come from using DRSs to build configurations are discussed below.

### **3.5 Path-Protecting $p$ -Cycle Classes**

Depending on the assumptions made about the transport network equipment, path protection  $p$ -cycle designs can fall into several categories. These are based on the types of working path/cycle relationships allowed in the final design and are introduced below, from the most specific to the most general. While the differences

between the categories provided below are subtle, they are provided to facilitate comparison between different design methods, as well as to explore the types of configurations encountered in this thesis.

### 3.5.1 Pure FIPP

This class of path-protecting  $p$ -cycles is one where all configurations contain a cycle and a set of protected mutually span-disjoint routes that are referred collectively to as a DRS. No working path is allowed to be in a z-case relationship with any cycle protecting it. An example of a pure FIPP configuration is illustrated in Figure 3.4. Any single failure will only affect at most one route per configuration (be it on-cycle or straddler) and the protection path provided by the cycle is unique for every working path. The protection path used to recover from failure is the same regardless of where the working path is affected. This category is fully failure independent and is useful under the assumption that nodal equipment is only capable of simple switching actions. This category of FIPP  $p$ -cycles is used extensively in Chapter 6 and Chapter 7.

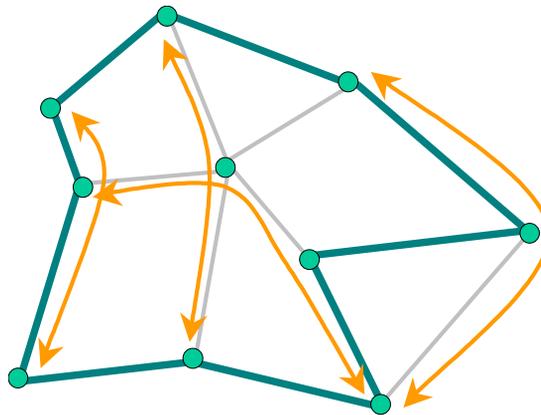


FIGURE 3.4. An example of a pure FIPP  $p$ -cycle configuration.

### 3.5.2 FIPP with z-Cases

This category is practically identical to the Pure FIPP class as it only allows for configurations to protect DRSs. The main difference between the two is that working routes are allowed to be in a z-case relationship with cycles protecting them. Allowing z-cases introduces a small amount of failure dependence to this class of FIPP  $p$ -cycles. As will be seen in Chapter 6, the difference between FIPP  $p$ -cycles with and without z-cases is very small in terms of capacity use (<10%), and z-cases are not very common. This category of FIPP  $p$ -cycles is encountered in Chapter 4, Chapter 5 and Chapter 6 of this thesis and is the most widely encountered FIPP category in literature.

### 3.5.3 Non-DRS FIPP

In this category of path-protecting  $p$ -cycles, configurations are not restricted to containing only mutually span disjoint routes. Routes in configurations may be non-disjoint as long as the protection paths used to recover these routes are. The term DRS now only applies in special cases where route sets happen to be mutually disjoint and in general the term path protected set (PPS) is used instead. As an example, two different configurations where non-disjoint paths are present are illustrated in Figure 3.5. Figure 3.5a shows two non-disjoint paths in a configuration that is 100% single failure restorable. Their restoration actions are illustrated by Figure 3.5b where both working paths are restored using mutually span disjoint protection paths. Figure 3.5c, shows a configuration where two working paths are non-disjoint, but cannot be protected if they both fail simultaneously because their protection paths are mutually non-disjoint and compete for spare capacity on the cycle as shown in Figure 3.5d. This category of FIPP  $p$ -cycles is encountered in Chapter 4.

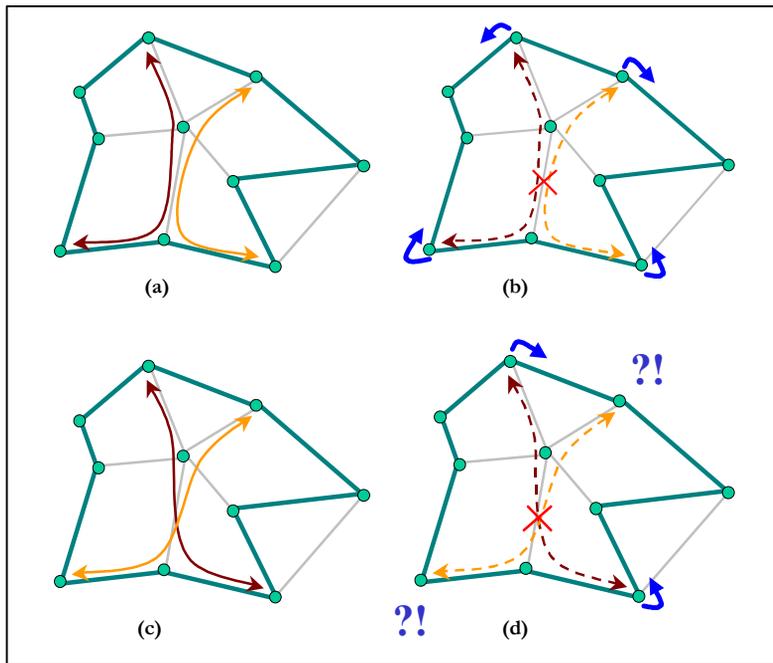
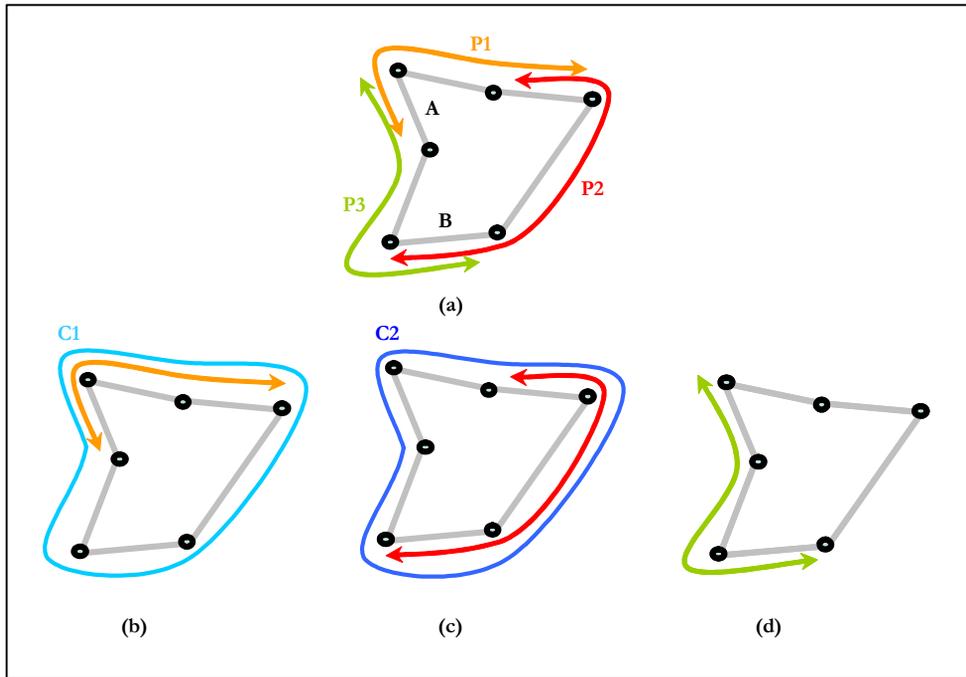


FIGURE 3.5. Non-disjoint working paths sharing the same cycle for protection. a) Valid case: pre-failure state. b) Both working paths are restored. c) Invalid case: pre-failure state. d) Only one path restored because the protection paths provided by the cycle are non-disjoint.

### 3.5.4 Generalized Path-Protecting $p$ -Cycles

The last category is the most general type of path-protecting  $p$ -cycle. Under generalized path-protecting (GPP)  $p$ -cycles, working paths can be protected using multiple protection paths as part of different configurations depending on where the working paths are affected by failure. For example consider Figure 3.6a where three unit paths  $P1$ ,  $P2$  and  $P3$  are shown. In order to protect these paths using any of the three categories provided above, three cycles would have to be placed. If any of the paths are placed together in the same configuration, the design would not be 100% single failure restorable because the working paths are non-disjoint and the same can be said for their protection paths. However, in this category it is possible to protect the same path set using only two cycles. The reason for this is because this category allows for a single working path to be protected using different cycles. Consider the scenario

where  $P1$  is protected using cycle  $C1$  as in Figure 3.6b,  $P2$  is protected using  $C2$  as in Figure 3.6c and  $P3$  is left unassigned, shown in Figure 3.6d. If span  $A$  fails,  $P1$  and  $P3$  are affected.  $P1$  is simply restored using  $C1$  because they are in the same configuration, and  $P3$  can be restored using  $C2$ . However, if span  $B$  fails then  $P2$  and  $P3$  are affected.  $P2$  can be restored using  $C2$ , while  $C1$  remains the only choice for  $P3$  to be restored.  $P3$  can be restored using either cycle, and the cycle that is actually used is dependant on where the network is affected by failure.



**FIGURE 3.6. Path-protecting  $p$ -cycle example. a) Path set to be protected. b,c) Two cycles  $C1$  and  $C2$  assigned to protect the first two paths  $P1$  and  $P2$  respectively. c)  $P3$  cannot be assigned a cycle for protection prior to failure.**

This category is the most flexible in terms of the ways that working paths are allowed to relate to one another and the cycles protecting them and yields the lowest bound on capacity in the class of path-protecting  $p$ -cycles. It is failure dependant and the knowledge of the exact location of failure is required for the correct restoration action to take place. GPP  $p$ -cycles are included here for completeness and to give the

reader a perspective on the relationships possible between working paths and cycles in a path-protecting  $p$ -cycle architecture. It is also intended to illustrate that FIPP  $p$ -cycles, whatever the flavor, are simply a special case of GPP  $p$ -cycles.

### 3.6 Prior Literature

The original work outlining the FIPP  $p$ -cycle concept is found in [38] and this work is expanded on in thorough detail in [39]. Additionally, [39] also introduces an ILP model called FIPP spare capacity placement (SCP) for generating FIPP  $p$ -cycle designs. Using a pre-generated cycle set and least cost working routing, the method attempts to optimally find a set of cycles to protect a set of demand relations against single span failures. The ILP model proposed in [39], however, is too difficult to solve even for medium sized networks, and a more practical method (called the FIPP DRS method) is introduced in [40]. This method focuses on pre-generating DRSs from a set of lowest cost working routes, enumerating eligible cycles to protect them and then using an ILP model to optimally decide which DRSs protected by which cycles should make up the final FIPP design. Given that this method is used extensively throughout this thesis, it is fully introduced in Chapter 4.

The FIPP DRS method is extended in [61] to joint capacity placement (JCP) where the DRSs are generated from a working route set which contains several routing options for each demand. This method, referred to as FIPP DRS JCP, is the main subject of Chapter 5.

Given that the problem of solving FIPP  $p$ -cycles grows very quickly with bigger networks, the OR technique of column generation (CG) is considered for FIPP design in [62]. This method allows the solver, starting with an initial set of configurations, to only generate new configurations when needed and only if adding them to the solution space results in a reduction of the design's overall cost. This technique, called FIPP CG, is presented in detail in Chapter 4. Additionally, a basic iterative heu-

ristic (IH) method referred to as FIPP IH is also introduced in [62] for comparison purposes as well as to generate the initial set of configurations for the FIPP CG method. It is also presented in Chapter 4. FIPP IH generates a single configuration per iteration by examining the cycles in the network and generating a DRS for each one using an ILP such that a credit score is maximized. The configuration with the highest score is selected to become part of the solution and the algorithm repeats until all the demands are protected.

The FIPP IH method is extended to joint capacity planning in [63] where the FIPP iterative joint design (IJD) method is proposed. Instead of using an ILP to generate DRSs for cycles, the FIPP IJD method utilizes two algorithms called difficult share first (DFS) and full protection (FP). DFS is a simple algorithm that generates DRSs by selecting first the working routes that are the longest and whose end-nodes have the highest capacity requests. The FP algorithm attempts to give a fairer representation to every demand by making sure that every eligible demand is protected as a part of a DRS at least once.

In [41] the genetic algorithm (GA) methodology is applied in order to generate optimal or near optimal GPP  $p$ -cycle solutions for larger network sizes. It is possible to extract FIPP  $p$ -cycle solutions from the GPP solutions using additional cycles whenever the constraint of failure independence is violated. The method works by using the proposed GPP  $p$ -cycle ILP as a fitness function and applying the GA steps of encoding, evaluation, selection and cross-over and finally mutation to the GPP design problem.

FIPP  $p$ -cycles were included in a set of transport networking case studies involving dual-failure analysis, minimum wavelength assignment, reach limit analysis under specific network equipment constraints, node failure analysis and cost optimization. These studies were done as part of the High AVAilability Network Architectures (HAVANA) project and mainly focused on investigation of pre-connected

network survivability architectures in the context of optical transparent networking. The results obtained from these studies can be found in [64], [65], [66] and [67]. These studies are covered in thorough detail in Chapter 6 and Chapter 7 of this thesis.

### 3.7 Summary

The FIPP  $p$ -cycle concept was introduced in this chapter. In summary, it is a path-protecting survivable networking architecture that has the following important properties:

- It is highly capacity efficient, reaching capacity efficiencies of shared-mesh schemes. A cycle is able to provide up to two protection paths for any demand that it protects as a straddler and is able to provide a single protection path for on-cycle demands.
- The protection paths are known well in advance and no cross-connection is required in real-time to form protection paths after the onset of failure. The paths are in a pre-connected, tested and ready state before failure happens.
- Only the end-nodes are responsible for failure recovery and the restoration actions taken to restore demands are typically fully failure location independent.
- The number of protection structures in a typical design is relatively small, easily visualized and exhibit low operational complexity.

## 4.1 Introduction

The main contribution of the following chapter is the formulation of a column generation (CG) method for FIPP  $p$ -cycle network design. This method is capable of generating optimal or near optimal solutions for network sizes where high quality solutions are difficult to obtain using other state of the art FIPP design methods. It does so by splitting the problem into a master and a pricing problem and solving these iteratively until an optimal solution is reached. This method also considers a more general configuration type than what has been considered in literature thus far. The above properties allow the CG method to generate highly efficient capacity solutions which exhibit a reduction in spare capacity usage of up to 36% relative to the solutions obtained using the most prevalent method in literature: the DRS method. This chapter also introduces a new iterative heuristic (IH) method for generating FIPP  $p$ -cycle designs. Although it was initially meant to only provide starting feasible solutions for the CG method, the results show that it is comparable in spare capacity to the solutions obtained using the DRS method.

In this chapter, the concept of column generation is introduced first. This is followed by the formulations of the CG and IH methods. The DRS method is also introduced in this chapter for completeness. This is followed by the description of the experimental setup and is concluded by the results and a short summary. This chapter is based on research done as part of a collaboration between the author and Dr. Wayne. D. Grover from TRILabs, University of Alberta and Dr. Brigitte Jaumard from Concordia University and Caroline Rocha from Universite de Montreal. The methods and the results found in this chapter were published in [62].

## 4.2 Motivation for Column Generation

The FIPP  $p$ -cycle concept is still fairly new and up to this point only a handful of methods for generating FIPP  $p$ -cycle designs have been developed. The original work on FIPP  $p$ -cycles introduced an ILP model [39], but this model proved to be difficult to solve. Other strategies, such as the DRS method [40] as well as the IH method (both covered in close detail later in this chapter) have mainly focused on the task generating FIPP  $p$ -cycle designs for practically sized networks, where the original ILP formulation failed to yield solutions. The same can be said about the joint capacity placement methods in [61] and [63]. One of the ongoing challenges in FIPP  $p$ -cycle design is the development of methods that are able to generate optimal or near optimal solutions in networks of practical size.

In practice, where the goal may be to generate good quality designs under a stipulated time limit, access to optimal solutions may not be essential. Academically, especially considering how little research has been done on FIPP  $p$ -cycles to date, optimal solutions may provide useful insights into the properties of highly efficient FIPP  $p$ -cycle designs. Furthermore, optimal solutions would allow us to explore the theoretical limits of capacity efficiency in FIPP  $p$ -cycle networks and would allow us to better compare this architecture to other closely related architectures such as SBPP. The insights gained can then be used to approach the design of heuristic methods; the quality of which may be graded by the efficiency of the solutions generated by these heuristics with respect to optimal FIPP  $p$ -cycle solutions. With this in mind, the CG approach is a promising strategy to improve the solution quality of FIPP  $p$ -cycle network design, while still achieving this in a reasonable amount of time. To this end, in this chapter we present a CG based method for generating FIPP  $p$ -cycle designs and contrast the results obtained against common FIPP  $p$ -cycle methods. Note that the emphasis in this chapter is not on achieving a speed-up in runtimes, but to achieve optimal or near optimal solutions while keeping the run-time needed for generating these solutions on par with previously proposed methods.

### 4.3 FIPP $p$ -Cycle Models

Three different models are presented in this chapter: the main new method called FIPP CG that makes use of column generation, an iterative heuristic method called FIPP IH and the FIPP DRS method, which is used extensively in this thesis. All three are SCP methods, meaning that the demands are routed using lowest cost routing and the spare capacity structures are generated afterward. Both FIPP DRS and FIPP IH methods are constrained to using DRSs as part of their configurations and this was done in order to reduce the complexity of the problem. The new CG method (because of how the problem is formulated) lifts this constraint and allows for non-disjoint working paths to be protected by the same cycle. Furthermore, where both FIPP DRS and FIPP IH methods allow for  $z$ -case cycle/working path relationships to occur (refer to Section 3.3.3 for more details regarding the  $z$ -case), the formulation of the FIPP CG method does not.

The models provided in this section share some set and parameter definitions. In order to avoid multiple declarations, these sets and parameters are presented below.

#### **Sets:**

$S$             Set containing the spans of the network.

$D$             Set of demand relations.

$P$             Set of  $p$ -cycles.

#### **Parameters:**

$d^r$             This parameter contains the number of unit demands for demand relation  $r$ .  $r \in D$ .

$c^i$	This parameter corresponds to the cost per unit of channel capacity on span $i$ . $i \in S$ .
$\pi^{p,i}$	Equal to 1 if cycle $p$ crosses span $i$ and is 0 otherwise. $p \in P, i \in S$ .
$\pi^{r,i}$	Equal to 1 if demand $r$ crosses span $i$ and is 0 otherwise. $r \in D, i \in S$ .
$x^{p,r}$	Equal to 1 if demand $r$ is in an on-cycle relationship with cycle $p$ , equal to 2 if demand $r$ is in a straddling relationship with cycle $p$ , and is equal to 0 otherwise. $p \in P, r \in D$ .

#### 4.4 FIPP $p$ -Cycle Column Generation Method

Column generation techniques [45] can be employed to solve linear programs with a very large number of variables and where the constraints can be expressed implicitly. They allow for the decomposition of the initial linear problem into two distinct parts: the master problem and the pricing problem. The master problem corresponds to a linear program with explicit constraints (such as variable bounds) and implicit constraints (such as constraints governing how the variables interrelate) expressed in the properties of the coefficients of the constraint matrix. The pricing problem is defined by the optimization of the reduced cost subject to the set of implicit constraints. The aim of the pricing problem is to help identify the most useful columns of the master problem.

Column generation is closely related to the simplex method and, like the simplex method, it is an iterative procedure. Where the simplex method selects a new column with a negative reduced cost to enter the basis, column generation uses the pricing problem to generate said column in an effort to improve the value of the

objective function of the master problem. Any column found by the pricing problem with a negative reduced cost entails an improvement in the objective value of the master problem, otherwise, if no such column is found, it can be concluded that the current solution is indeed optimal. Column generation can be combined with branch-and-bound techniques for solving ILPs with a large number of variables [68]. Branching rules have to be devised properly in order to prevent a large number of sub-problems from forming in the search tree associated with the branch-and-bound. This can be done by either branching on the variables of the master problem using cuts, or by branching on the variables of the pricing problem using classical branching schemes.

#### **4.4.1 FIPP $p$ -Cycle Column Generation ILP**

The following subsection introduces the FIPP  $p$ -cycle column generation (CG) model. In the master problem presented below, each variable corresponds to a configuration. Recall that a configuration refers to unit cycle and a set of demands relations protected by it. Under column generation, configurations are not precomputed but iteratively generated as part of the pricing problem. Their reduced cost is used in order to identify the configurations that allow for the largest reduction in the master problem's objective value.

A column generation always corresponds to a decomposition of the set of constraints between the master problem and the pricing problem. In this case, the master problem (Section 4.4.1.1) includes demand related constraints that link the configurations into a solution. The pricing problem (Section 4.4.1.2) contains the constraints that are associated with a specific configuration.

##### ***4.4.1.1 FIPP CG Master Problem***

The FIPP  $p$ -cycle column generation master problem is formulated as follows:

**Sets:**

$G$  Set of all configurations.

$P^g$  Unit set containing the  $p$ -cycle generated for configuration  $g$ .

$$P^g \subseteq P, g \in G.$$

**Parameters:**

$COST^g$  This parameter corresponds to the cost of configuration  $g$ .

$$COST^g = \sum_{i \in S} \pi^{p,i} \cdot c^i \quad p \in P^g, g \in G.$$

$a^{g,r}$  This parameter corresponds to the number of protection units provided by cycle  $P^g$  for demand relation  $r$  as part of configuration

$$g. a^{g,r} \in \{0, 1, \dots, d^r\}, r \in D, g \in G.$$

$a^g$  This is a vector representing configuration  $g$ .  $a^g = (a^{g,r})_{r \in D}$ ,  
 $g \in G$ .

**Variables:**

$z^g$  Integer variable corresponding to the number of copies of configuration  $g$  used in the solution.  $z^g \geq 0, g \in G$ .

**FIPP CG Master:**

$$\text{Minimize: } \sum_{g \in G} COST^g \cdot z^g \quad (4.1)$$

**Constraints:**

$$\sum_{g \in G} a^{g,r} \cdot z^g \geq d^r \quad \forall r \in D \quad (4.2)$$

The objective function (4.1) minimizes the total spare capacity while constraint (4.2) ensures every demand request is satisfied. Note that in order to reduce the number of configurations, only maximal configurations are considered. A configuration  $g$  is maximal if there does not exist another configuration  $g'$  such that  $a^{g'} \geq a^g$ . The use of maximal configurations allows for instances where some demands may be over satisfied. However, this only occurs when doing so does not require any additional cost compared to satisfying the demand exactly.

#### 4.4.1.2 FIPP CG Pricing Problem

By definition, the pricing problem corresponds to the optimization problem of minimizing the reduced cost (with respect to the linear programming definition) subject to the constraints that must be satisfied by a given configuration. The constraints are responsible for defining the configuration cycle, identifying which demand relations are to be included in the configuration and for prohibiting a span from being used as a working and a protection span at the same time for the same demand. The FIPP  $p$ -cycle CG pricing problem uses the following formulation as well as additional notation:

##### Sets:

$N$  Set containing the nodes of the network.

$N^r$  Set of end-nodes  $e^{r,1}$  and  $e^{r,2}$  for demand relation  $r$ .

$$\{e^{r,1}, e^{r,2}\} \in N^r, N^r \subseteq N, r \in D.$$

$S^n$  Set of spans adjacent to node  $n$ .  $S^n \subseteq S, n \in N$ .

$\Psi(N')$  Set of all spans whose end-nodes belong to  $N'$ .  $N' \subseteq N$ .

$\Omega(N')$  Set of all spans whose one end-node belong to  $N'$  and the other does not.  $N' \subseteq N$ .

**Parameters:**

$u^r$  This parameter corresponds to the dual price (shadow price).  $r \in D$ .

$\rho^{r,r'}$  Equal to 1 if demands  $r$  and  $r'$  are non-disjoint and is 0 otherwise.  
 $r \in D, r' \in D$

**Variables:**

$x^i$  Equal to 1 if the cycle being generated crosses span  $i$  and is 0 otherwise.  $i \in S$ .

$x^{r,i}$  Equal to 1 if any protection path for demand  $r$  uses span  $i$ .  
 $r \in D, i \in S$

**FIPP CG Pricing:**

$$\text{Minimize: } \overline{COST^g} \tag{4.3}$$

Where:

$$\overline{COST^g} = COST^g - \sum_{r \in D} a^{g,r} \cdot u^r = \sum_{i \in S} x^i \cdot c^i - \sum_{r \in D} \sum_{i \in S^{e^{r,1}}} x^{r,i} \cdot u^r \tag{4.4}$$

**Constraints:**

$$\sum_{i \in S^n} x^i \leq 2 \quad \forall n \in N \quad (4.5)$$

$$\sum_{i' \in S^n, i' \neq i} x^{i'} \geq x^i \quad \forall n \in N, \forall i \in S^n \quad (4.6)$$

$$\sum_{i \in \Omega(N')} x^i \geq x^{i'} + x^{i''} - 1 \quad N' \subseteq N, \forall i' \in \Psi(N'), \forall i'' \notin \Psi(N'') \quad (4.7)$$

$$x^i \geq x^{r,i} \quad \forall i \in S, \forall r \in D \quad (4.8)$$

$$\sum_{i \in S^{e^{r,1}}} x^{r,i} = \sum_{i \in S^{e^{r,2}}} x^{r,i} \quad \forall r \in D \quad (4.9)$$

$$\sum_{i \in S^n} x^{r,i} \leq 2 \quad \forall n \in N, \forall r \in D \quad (4.10)$$

$$\sum_{i' \in S^n, i' \neq i} x^{r,i'} \geq x^{r,i} \quad \forall r \in D, \forall n \in N \setminus \{e^{r,1}, e^{r,2}\}, \forall i \in S^n \quad (4.11)$$

$$2 - x^{r,i} - x^{r,i'} \geq \rho^{r,r'} \quad \forall i \in S, \forall (r, r') \in D^2, r \neq r' \quad (4.12)$$

$$x^{r,i} \leq 1 - \pi^{r,i} \quad \forall i \in S, \forall r \in D \quad (4.13)$$

The objective function is given by (4.3) and expanded in (4.4) and corresponds to the reduced cost of a column of the master problem. The expression of the objective function corresponds to the cost of the cycle less the prices of the protected demands. The first three constraints ((4.5)(4.6)(4.7)) take care of the construction of the cycles. Constraints (4.5) and (4.6) ensure that every node either has 0 or 2 adjacent spans used by the cycle being formed. Constraint (4.7) prevents sub-cycles from forming. In other words, this constraint makes sure that only one cycle is formed for the configuration. This constraint is a variant of the classical sub-cycle elimination constraint of the Traveling Salesman Problem (TSP), with the difference being 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 sub-cycle elimination constraints. However, 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, the pricing problem may need only be solved exactly once to confirm that the heuristic has indeed found the optimal solution.

Constraint (4.8) establishes the relationship between the cycle and the protection paths provided to the demand relations in the configuration. It makes sure that every protection path crosses the same spans as the cycle. Constraints (4.9),(4.10) and (4.11) correspond to the definition of the protection paths. 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 provided by the cycle under construction. Constraint (4.9) states that the protection flow exchanged between the end-nodes must be equal or, in other words, that the protection path must end at both end-nodes of a demand. At the end-nodes, constraint (4.10) allows at most two protection paths for each demand relation. At intermediate nodes, constraints (4.10) and (4.11) together ensure flow conservation. They make sure that the number of incoming/outgoing of flows at these nodes is either 2 or 0, corresponding to whether the intermediate node is crossed by the protection path or not, respectively. Constraint (4.12) makes sure that protection paths of non-disjointly routed demands do not cross the same spans. Constraint (4.13) prevents the same span from being used by the working route of a demand relation and its protection path simultaneously.

## 4.5 FIPP $p$ -Cycle Iterative Heuristic Method

The FIPP iterative heuristic (IH) method is a new basic method for generating FIPP  $p$ -cycle designs. It is introduced here as a supplementary method to the FIPP CG method. The main goal of this method is to generate reasonably efficient FIPP  $p$ -cycle solutions that can be used as input to the CG method, with the overall aim of

shortening the time required for the CG method to arrive at an optimal solution. It works by solving a sub-problem, called FIPP IH ILP 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 configuration (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 as part of this configuration is removed from the demand set. The FIPP IH ILP SubProblem is re-solved for every cycle in the network using the new remaining demand set. This process continues until there are no demands left to protect. Figure 4.1 summarizes the algorithm.

```

FIPP_IH {
  repeat {
    for every cycle  $p$  do {
      solve FIPP_IH_ILP_SubProblem;
    }
    Select the configuration with the highest score;
    Record the configuration as part of the solution;
    Remove the units demands protected by the configuration
    from the set of remaining unprotected demands;
  } while the set of remaining unprotected demands is not empty;
}

```

FIGURE 4.1. The FIPP IH algorithm.

#### 4.5.1 FIPP $p$ -Cycle Iterative Heuristic ILP SubProblem

The FIPP IH ILP SubProblem is formulated as follows:

**Sets:**

$D^*$  Set of non-zero demand relations.

**Parameters:**

$c^p$  This parameter corresponds to the cost of cycle  $p$ .

$$c^p = \sum_{i \in S} c^i \cdot \pi^{p,i}, p \in P.$$

**Variables:**

$n^{p,r}$  Equal to 1 if cycle  $p$  protects demand  $r$  and is 0 otherwise.  
 $p \in P, r \in D$ .

**FIPP IH ILP SubProblem:**

$$\text{Maximize: } \sum_{r \in D^*} \frac{x^{p,r} \cdot n^{p,r}}{c^p} \tag{4.14}$$

**Constraints:**

$$\sum_{r \in D^*} \pi^{r,i} \cdot n^{p,r} \leq 1 \quad \forall i \in S \tag{4.15}$$

The objective function (4.14) 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 per iteration. The cycle with the highest credit score is the cycle that gets added to the solution. Constraint (4.15) ensures that the working routes sharing the same cycle for protection are mutually span disjoint.

## 4.6 FIPP $p$ -Cycle DRS method

The FIPP disjoint route set (DRS) method [40] is an extensively used method in this thesis as well as in literature and is presented here for reference. The results obtained using this method are contrasted with the results obtained by FIPP CG and

FIPP IH in the results section (Section 4.9) of this chapter. The FIPP DRS method is illustrated in Figure 4.2. It works by generating working routes for every demand in the network and then pooling all these working routes into an *eligible route set*. A single DRS is formed by first picking a *seed* route for the DRS and then randomly picking routes from the eligible route set and checking whether they are disjoint from the routes already in the DRS or not. If the picked route is disjoint from all the routes already in the DRS, then it is added to the DRS and is ignored otherwise. The eligible route set is polled until the DRS reaches a specified size, which ranges from one route to a user specified maximum, controlled by the *max DRS size* parameter. Allowing for a variety of DRS sizes allows the solver to assemble a more efficient solution [69]. The *number of DRSs per demand* parameter controls how many times a particular route appears as a *seed* for a DRS. Particular care is taken to make sure that every demand appears in a single route DRS in addition to the DRSs generated using the method above. This is to allow the solver to choose from this set of DRSs when protection is needed for single highly-sized demands.

Once the DRSs are generated, several cycles, the number of which is specified by the *eligible cycles per DRS* parameter, are assigned to every DRS. The cycles selected are the shortest distinct cycles possible that cross every node that is an end-node to the routes in the DRS being considered. Making sure that every end-node is crossed guarantees that the cycle can protect the DRS. The DRSs and the cycles assigned to protect them are then passed as parameters to the solver, which selects the least cost combination of DRSs and cycles.

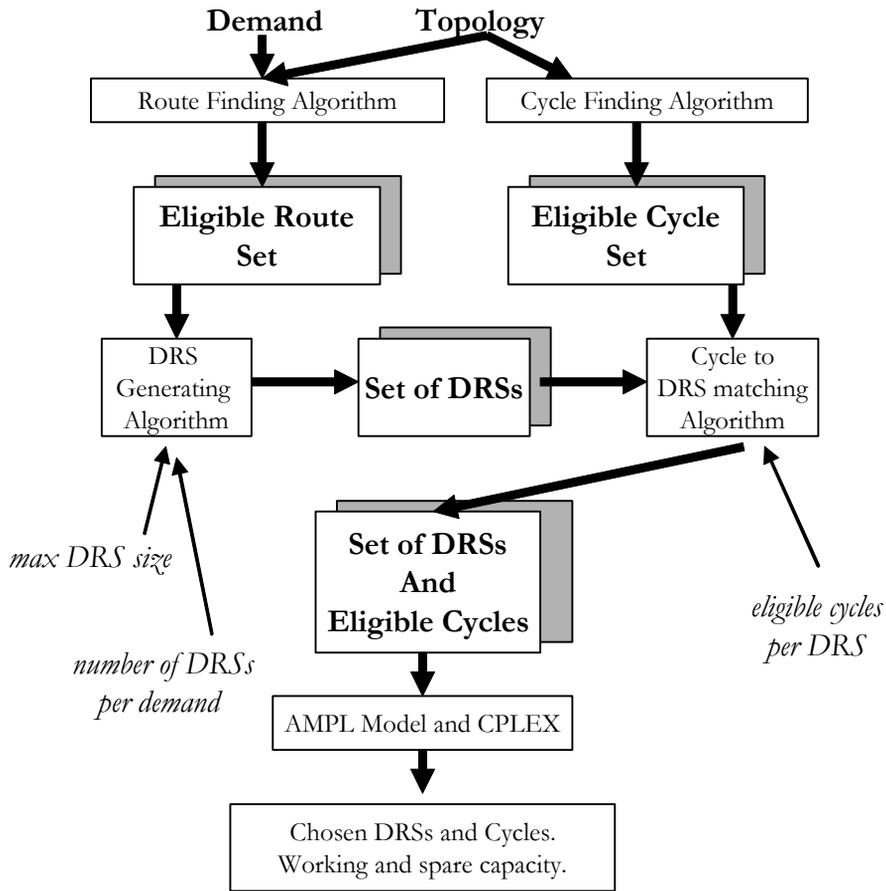


FIGURE 4.2. FIPP  $p$ -cycle DRS method.

#### 4.6.1 FIPP $p$ -Cycle DRS ILP

The FIPP  $p$ -Cycle DRS ILP is formulated as follows:

**Sets:**

$C$  Set containing all eligible DRSs.

$C^r$  Set of all DRSs containing the working route belonging to demand relation  $r$ .  $C^r \subseteq C, r \in D$ .

$P^c$  Set of eligible  $p$ -cycles that provide protection for DRS  $c$ .

$$P^c \subseteq P, c \in C.$$

**Variables:**

$s^i$  Integer variable corresponding to the total spare capacity placed on span  $i$ .  $s^i \geq 0, i \in S$ .

$n^{p,c}$  Integer variable corresponding to the number of unit capacity copies of cycle  $p$  used to protect all instances of DRS  $c$ .

$$n^{p,c} \geq 0, p \in P^c, c \in C.$$

$n^p$  Integer variable corresponding to the total number of unit capacity copies of cycle  $p$  used.  $n^p \geq 0, p \in P$ .

**FIPP DRS ILP:**

$$\text{Minimize: } \sum_{i \in S} c^i \cdot s^i \quad (4.16)$$

**Constraints:**

$$\sum_{c \in C^r} \sum_{p \in P^c} x^{p,r} \cdot n^{p,c} \geq d^r \quad \forall r \in D \quad (4.17)$$

$$n^p = \sum_{c \in C} n^{p,c} \quad \forall p \in P \quad (4.18)$$

$$s^i \geq \sum_{p \in P} n^p \cdot \pi^{p,i} \quad \forall i \in S \quad (4.19)$$

The objective function (4.16) minimizes the total cost of spare capacity needed to protect the demands specified. Constraint (4.17) places enough FIPP  $p$ -cycles to protect the specified demands. Constraint (4.18) calculates the total number of each cycle needed and constraint (4.19) places enough spare capacity to support the cycles selected.

## 4.7 Solving the CG model

Now that the ILP models have been presented, some issues regarding the FIPP CG method can be addressed. One of the main difficulties with the CG method lies in solving the pricing problem due to the exponential number of constraints required to prevent the formation of sub-cycles. However, as mentioned before, the pricing problem does not need to be solved exactly at each iteration of the CG process as long as we are able to find a configuration with a negative reduced cost. The pricing problem can be described as a multi-commodity flow problem with side constraints corresponding to the definition of a cycle on which the flows circulate. Constraints in constraint set (4.7) that prevent sub-cycles from being considered are very costly as there are an exponential number of them. In order to overcome this difficulty, they are introduced only as needed following the principle of the “lazy constraints” feature of CPLEX 10.0 solver. It means that the pricing problem is solved iteratively, starting with no (or a very small number of) sub-cycle constraints. Additional sub-cycle constraints are added only as needed until a feasible solution is reached. In order to save time, the solver parameters were set so that the solver returned the first feasible solution to the pricing problem that had a negative reduced cost. In practice, we need to introduce a very small number of sub-cycle constraints and therefore the computing cost of the iterative process is counter-balanced by the smaller sizes of the pricing problems that need to be solved. In fact, the pricing problem very rarely needed to be solved more than twice. Note that no sub-cycle constraints are eliminated after they have been introduced. Meaning that their number increases as the solver proceeds fur-

ther down the branch-and-bound search tree. It was found that at each iteration, most of the time the solution of the first pricing problem was very often feasible and satisfied all sub-cycle constraints even if only a small number of them had been explicitly introduced in the pricing problem.

It turns out that only a limited number of columns were needed to be generated until the optimality condition of the linear programming relaxation (when no more columns with negative cost could be found) was reached. For this reason, no custom branch-and-bound (B&B) method was developed to get an integer solution for the master problem. Instead the CPLEX 10.0 B&B feature was used to solve the restricted master problem. While not fully optimal in some cases, the solutions obtained were close to optimality and quite satisfactory in comparison with those obtained previous FIPP models, as will be shown in Section 4.9.

## 4.8 Experimental Setup

The solutions for the FIPP CG, FIPP IH and FIPP DRS design models were obtained by implementing the prior model in C++ and the latter two models in AMPL 9.0. The FIPP IH and FIPP 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 [11] containing 11 nodes and 26 spans as well as a 15-node family of networks with spans varying from 16 to 30 [70]. The COST 239 network represents a European fiber optic network topology, while the 15-node family of networks was designed as part of the research done by our group (Network Systems) at TRILabs. This family was designed to be characteristic of real transport networks in a sense that each member of the family is planar or near planar and exhibits high local nodal connectivity (nodes generally connect to nodes in their vicinity). Additionally, the nodal degree of any individual node does not exceed twice the average nodal degree of the network. The 15-node family was designed by

starting out with a 15-node topology where the average nodal degree was 4.0 and then successively removing spans one at a time while retaining the property of bi-connectivity and keeping all the nodal positions the same across all members. The overall goal of this network family was to allow investigations of various transport networking architectures and metrics (such as dual-failure restorability) with respect to changes in nodal degree.

The number of demand relations is 55 for the COST239 instance, and 105 for all instances of the 15-node family. The details regarding both of these can be found in the Appendix (Section 10.1 and Section 10.3). For the FIPP DRS solutions provided, 30 DRSs were generated per demand in the network. This means that every demand appeared in at least 30 different DRSs. For every DRS, 3 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 [40], 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 IH solutions were obtained by running the FIPP IH algorithm where all possible cycles in the network were considered as eligible candidates.

## 4.9 Results

The FIPP CG method uses the results obtained by the FIPP IH method as starting feasible solutions. The results for the FIPP CG method are summarized in Table 4.1. The table reports the final cost of spare capacity as well as the gap against optimality which is the gap between the optimal solution of the linear relaxation of the master problem and the best known integer 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 -). The total number of unit

cycles as well as the number of distinct cycles used in the final design (of which there can be several instances of) are also provided in Table 4.1. Note that the total # of unit cycles column also corresponds to the # of unit configurations used in the solution. The overall number of generated configurations is provided using two columns: The first column corresponds to the number of generated configurations that had positive master variables  $z^g$  and the second column corresponds to the number configurations that had master variables set to null. The column where  $z^g > 0$  essentially refers to the number of distinct configurations used in the solution (of which there could be several instances of) while the  $z^g = 0$  column corresponds to the total number of configurations that were generated as part of the CG method, but which were not used in the FIPP CG solution. The second last column contains the number of distinct configurations that contain at least one pair of non-disjoint working paths. According to the table, approximately half of the used configurations contained non-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). Recall 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 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.

TABLE 4.1. Results obtained by the FIPP CG algorithm.

Problem Instance	Cost of Spare	Gap (%)	# Unit Cycles	# Distinct Cycles	Overall # of Configs.		# of Non-Disjoint Configs.	Overall # of Cycles
					$z^g > 0$	$z^g = 0$		
<b>COST239</b>	68840	2.5	17	13	15	534	6	286 (20)
<b>15n30s-16s</b>	387958	0.0	185	3	73	169	0	3 (3)
<b>15n30s-17s</b>	286901	0.2	140	6	72	217	18	7 (6)
<b>15n30s-18s</b>	266080	0.2	126	7	60	350	22	11 (10)
<b>15n30s-19s</b>	229854	0.1	108	11	59	338	20	21 (16)
<b>15n30s-20s</b>	217963	0.1	108	13	64	472	22	31 (23)
<b>15n30s-21s</b>	192269	0.4	101	17	63	412	26	50 (27)
<b>15n30s-22s</b>	182550	0.6	101	19	66	411	30	72 (34)
<b>15n30s-23s</b>	177432	0.5	100	27	61	505	32	111 (45)
<b>15n30s-24s</b>	175990	0.6-	100	29	58	636	23	178 (45)
<b>15n30s-25s</b>	157140	0.9-	92	36	56	504	21	196 (43)
<b>15n30s-26s</b>	106526	6.1-	55	39	45	395	18	287 (39)
<b>15n30s-27s</b>	117220	5.5-	61	41	45	399	28	287 (51)
<b>15n30s-28s</b>	114055	4.5-	59	44	46	333	23	303 (44)
<b>15n30s-29s</b>	106449	2.6-	56	33	42	534	28	345 (39)
<b>15n30s-30s</b>	110738	1.0-	62	40	51	473	34	326 (38)

The results for FIPP DRS and FIPP IH are documented in Table 4.2 and Table 4.3, respectively. These tables contain the final spare capacity 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 4.2 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 because of 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 reach-

ing 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 constraining the number of DRSs and cycles appearing in the data set are introduced, the DRS method may not yield an optimal solution. Instead, it is only capable of finding the optimal combination of the DRSs and cycles given to it as input.

**TABLE 4.2. FIPP DRS results.**

<b>Problem Instance</b>	<b>Cost of Spare</b>	<b># of Unit Cycles</b>	<b># of Distinct Cycles</b>	<b>Cost Relative to FIPP CG (%)</b>
<b>COST239</b>	93345	24	17	36
<b>15n30s-16s</b>	393094	189	3	1
<b>15n30s-17s</b>	305065	146	6	6
<b>15n30s-18s</b>	272661	127	8	2
<b>15n30s-19s</b>	246035	110	12	7
<b>15n30s-20s</b>	239942	113	17	10
<b>15n30s-21s</b>	208501	101	26	8
<b>15n30s-22s</b>	202771	101	40	11
<b>15n30s- &gt;22s</b>	-	-	-	-

As an alternative to the FIPP 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 4.3.

TABLE 4.3. FIPP IH results.

<b>Problem Instance</b>	<b>Cost of Spare</b>	<b># of Unit Cycles</b>	<b># of Distinct Cycles</b>	<b>Cost Relative to FIPP CG (%)</b>
<b>COST239</b>	94095	25	20	37
<b>15n30s-16s</b>	396935	192	3	2
<b>15n30s-17s</b>	310094	152	6	8
<b>15n30s-18s</b>	291047	139	10	9
<b>15n30s-19s</b>	259825	129	16	13
<b>15n30s-20s</b>	246505	124	23	13
<b>15n30s-21s</b>	215708	114	27	12
<b>15n30s-22s</b>	198789	104	34	9
<b>15n30s-23s</b>	199378	106	45	12
<b>15n30s-24s</b>	201207	107	45	14
<b>15n30s-25s</b>	179897	100	43	14
<b>15n30s-26s</b>	115848	60	39	9
<b>15n30s-27s</b>	137887	74	51	18
<b>15n30s-28s</b>	128957	71	44	13
<b>15n30s-29s</b>	126745	73	39	19
<b>15n30s-30s</b>	132294	79	38	19

The spare capacity cost of the CG method is lower than both the DRS and the IH method solutions. Moreover the DRS and the IH methods are relatively close to each other in cost. The cost improvement when comparing the IH and DRS results to the FIPP CG solutions is as much as 19% and 11% for the 15n30s and is 37% and 36% for the cost 239 network, respectively. 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 subset of all possible configurations and is thus 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 different cycles and their protected sets.

## 4.10 Conclusion

In this chapter we introduced and investigated a first column generation (CG) model for the efficient design of FIPP  $p$ -cycles. We compared it successfully against the FIPP DRS method as well as the new FIPP IH heuristic. While previous FIPP SCP methods work under the assumption that working paths are mutually disjoint, this is not the case for the FIPP CG model. 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 thus help with the invention and improvement (in terms of mean run time or solution quality, for example) of heuristic methods. It was not the goal of this study to focus on the mean running times of the algorithms used. Despite this, a time limit was imposed on the experiments to ensure that the solutions were obtained within a reasonable amount of time (5 hours). The main focus of this study was to obtain solutions of extremely high quality to serve as a lowest bound on the capacity use. Despite the quality of the solutions presented, there is still room for improvement of the CG model. In particular it could be sped up if a heuristic for the pricing problem, that is able to generate configurations with negative reduced cost, could be found.



## 5.1 Introduction

The main contribution of this chapter is the extension of a well known FIPP  $p$ -cycle DRS design method [40] from a spare capacity placement (SCP) paradigm, where protection structure placement is determined based on known working routing (typically routed using shortest hops or minimum cost), to a joint capacity placement (JCP) paradigm, where both spare and working capacity are placed simultaneously. The results obtained using the new FIPP DRS JCP method (which show an improvement over SCP solutions in the order of 13%) are first compared to JCP solutions of closely related architectures such as span-protecting  $p$ -cycles and SBPP. This is followed by a series of experiments, the goal of which is the investigation of exactly how FIPP  $p$ -cycle designs are improved by jointness. These investigations allow for a closer look into where the reduction in cost relative to SCP solutions comes from in FIPP  $p$ -cycle designs and allow for detection of threshold effects and factors limiting further improvements.

The chapter is organized as follows: First the concept of joint capacity design is introduced in detail. This is followed by a review of the formulation of the FIPP DRS JCP method and a detailed description of the experimental setup. Next, a preliminary solution using default parameter values is generated as a starting point. This is followed by the set of investigations where individual parameters are varied and their effects observed. The chapter is concluded with a look at a visual representation of the best JCP solution found and a brief summary. The work presented in this chapter is based on the collaborative work of the author with Dr. Adil Kodian and Dr. Wayne D. Grover. The publication of the methods presented and results obtained in this chapter can be found in [61].

## 5.2 Joint Optimization of Working and Spare Capacity

A common starting point for the design of transport networks under any architecture involves methods where the demands are routed along shortest/lowest cost routes and the spare capacity is placed to protect these routes with the overall goal of placing as little spare capacity as possible while still guaranteeing 100% restorability against single span failures. FIPP  $p$ -cycles are no exception to this approach, which is referred to as spare capacity placement (SCP): Both initial publications, namely FIPP ILP [39] and FIPP DRS [40], present SCP methods. There are, however, reasons to think that a jointly optimized design may be especially attractive in a network using FIPP  $p$ -cycle protection. Under a joint capacity placement (JCP) design model working routing, working capacity, spare capacity, and protection structure definitions are all decided concurrently, with the overall goal of minimizing the total capacity or cost of the network design. Essentially, under JCP the main inputs to the problem are the network topology and the demand matrix (specifying which demands relations must be serviced). This information is passed to the solver, that is then responsible for determining the working routing for the demand relations as well as the spare capacity placement required to protect these routes. The solver may generate these routes from scratch or choose from a set of pre-generated routing alternatives for each demand relation. In this chapter we define a JCP version of the FIPP  $p$ -cycle network design problem and study the properties of the resulting network designs to understand how and why joint design differs and improves on SCP solutions. This extension is motivated by the fact that, in general, JCP solutions are lower in total cost than their SCP counterparts, and because it turns out that for FIPP  $p$ -cycle designs obtained using the existing DRS-based approach, the benefits of jointness can be gained without overwhelming additional computational effort.

In SCP, working path routing decisions are independent of and precede the protection design problem. The resulting routings and working capacity accumulations then are inputs to the survivability design process. In contrast, the aim of JCP is to minimize the total cost of working and spare capacity at the same time. This type of

optimization aims to exploit how the choices of working routes can interact synergistically with the investment in spare capacity and protection structures that are shared over potential single-failure scenarios. To do this, JCP methods allow the working paths between node pairs to deviate from the strictly shortest routes (routes otherwise used in the corresponding SCP problem). This gives the solver the opportunity to consider working routing alternatives so that the overall effectiveness of spare capacity in protecting working capacity is maximized.

In other works on span restorable mesh networks [27][28] joint capacity placement was found to yield reductions up to 28% in total capacity relative to the non-joint designs. It has also been found in [9][69] that, using joint optimization, it was possible to achieve an improvement of 25% in span-protecting  $p$ -cycle networks. Inspection has shown, however, that the improved efficiency is typically not associated with any wild deviations from the shortest path routing. Rather, it arises mainly from providing the solver with the ability to make choices between nearly equal shortest working route alternatives where they exist. A finding from [27] is that working routes chosen in joint design are typically only a few percent longer than the shortest route in each case. In [9] (p. 314–316), it is explained that jointness typically works more through load-leveling effects among almost equal shortest routes and is unlikely to involve significantly longer routes that are somehow highly synergetic with the spare capacity placement. The above considerations motivate the extension of FIPP  $p$ -cycle network design to JCP, the method for which is introduced in the following section.

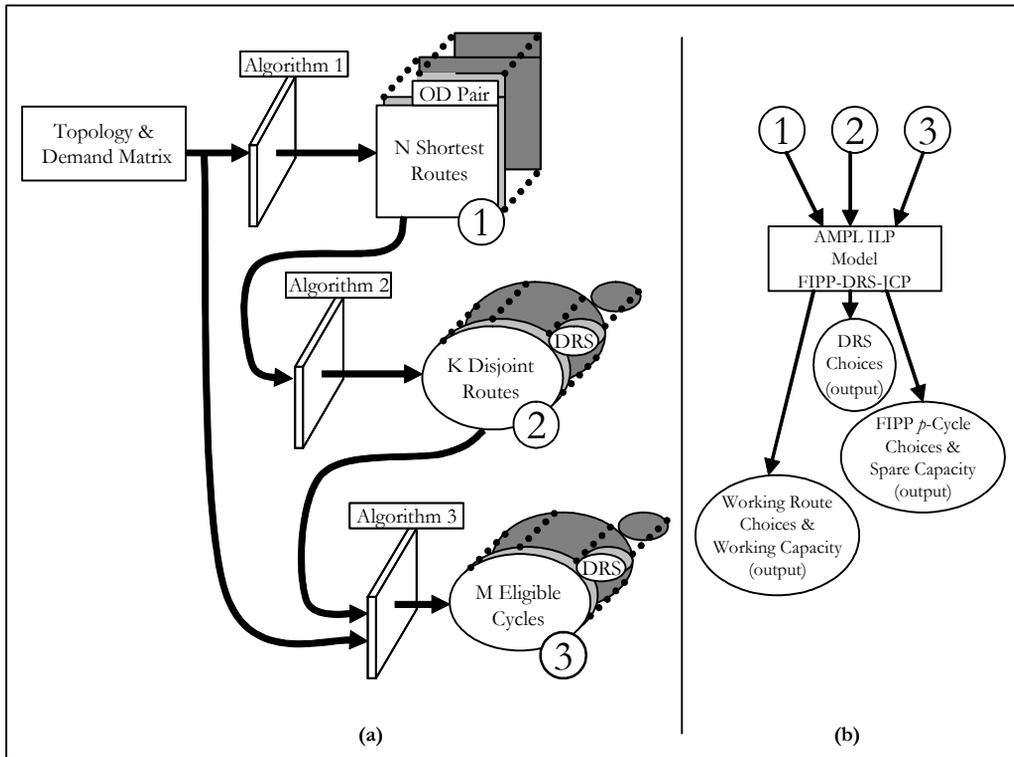
### 5.3 FIPP $p$ -Cycle DRS JCP Method

The overall strategy for the FIPP  $p$ -cycle JCP DRS design method that has been developed here is illustrated in Figure 5.1. For every node pair exchanging non-zero demand, *Algorithm 1* first finds  $N$  shortest routes between the origin and destination nodes of that demand. In this regard, setting  $N = 1$  reduces the JCP problem

to the SCP problem in [40]. The complete set of routes produced by this algorithm for all relations is referred to as the eligible route set. *Algorithm2* then creates eligible DRSs by combining selections of route choices that are disjoint from one another into sets. This is a modified version of the *GenerateDRS* algorithm from [40]. The pseudocode of the algorithm is found in Figure 5.2. Each candidate DRS is generated by first being seeded with the shortest route of some relation. (Later measures will ensure that a minimum number of DRSs  $m$  include the shortest route of that relation.) The DRS grows by a process where single routes are randomly picked from the eligible route set and are added to the DRS only if they meet the disjointness criteria with respect to the routes already in the DRS.

The DRSs grow until they reach the maximum number of routes in a DRS, set by the user, or until the algorithm is terminated using the *RandomExit* function (explained in detail below). Previous experience with heuristic design methods for  $p$ -cycles, such as Capacitated Iterative Design Algorithm (CIDA) [71], has shown that, in general, candidate structures (and, here by implication, candidate route set structures) should not include only the largest most potentially individually efficient candidates. By efficient, we mean cycles that have the relatively high ratios of the number of protected working paths to cycle cost. The most efficient structures using this metric tend to be large cycles that protect large DRSs with many straddling relationships. Rather, in these semi-heuristic ILP approaches, it is beneficial to provide the solver with a variety of candidate sizes. To have the same effect in enumeration of candidate DRSs, the *RandomExit* function is used in the algorithm in Figure 5.2. The effect is that it is increasingly likely to terminate the DRS formation loop as the candidate DRS gets bigger. The philosophy is that we recognize that, by themselves, the largest possible assemblies of mutually disjoint routes stand to become individually the most efficient single FIPP  $p$ -cycle structures. Following that logic alone, we would never stop adding to each DRS formation prematurely in this way, until further route addi-

tions become impossible for each candidate DRS. Instead, *RandomExit* assures that a variety of DRS sizes (in terms of number of routes included) is represented in the ILP problem instance that is being developed.



**FIGURE 5.1. a) The data set generation flow model for the FIPP DRS JCP problem: (1) Eligible shortest routes set per node pair, (2) Disjoint route set candidates, (3) Eligible cycles for each DRS candidate. b) The data set and the model are combined and passed to the solver to yield a solution (outputs).**

```

GenerateDRSs() {
  for each non-zero demand  $r$  do {
    for number of DRSs per demand needed  $m$  do {
      Add shortest route of demand  $r$  to DRS  $c$ ;

      while total number of routes in DRS  $c$  is  $< K$ , the max. number of routes per DRS {
        Randomly select a working route  $x$  from the Eligible Route Set;

        if  $x$  is disjoint from all routes in DRS  $c$  {
          Add working route  $x$  to DRS  $c$ ;
        }
        RandomExit();
      }
      Add DRS  $c$  to DRS_Set;
    }
  }
  Return DRS_Set;
}

```

FIGURE 5.2. **GenerateDRSs pseudocode.**

To recap, so far, multiple route choices have been identified for each demand relation (*Algorithm 1*), and a large number of DRS candidates (*Algorithm 2*) have been formed from these routes in a way that ensures every eligible route is present in at least  $m$  DRSs. *Algorithm 3* takes care of the remaining ingredient needed: the candidate  $p$ -cycles to be associated with each DRS. It generates a number of eligible cycles  $M$  for each DRS candidate. These are cycles that pass through all the end-nodes of the corresponding DRS and are organized by length in increasing order ( $M$  shortest eligible cycles). The eligible route set, all the DRS candidates and the set of their corresponding eligible FIPP  $p$ -cycle candidates are then used to populate an instance of the FIPP DRS JCP ILP problem. The ILP solver then chooses the best combination of working routes, DRSs, and FIPP  $p$ -cycles using the model in Section 5.3.1.

### 5.3.1 FIPP $p$ -Cycle DRS JCP ILP

The FIPP  $p$ -cycle DRS JCP ILP is formulated as follows:

**Sets:**

- $S$  Set containing the spans of the network.
- $D$  Set of demand relations.
- $C$  Set containing all eligible DRSs.
- $C^q$  Set of all DRSs containing a working route  $q$  belonging to demand relation  $r$ .  $C^q \subseteq C, q \in Q^r, r \in D$ .
- $P$  Set of eligible  $p$ -cycles.
- $P^c$  Set of eligible  $p$ -cycles that provide protection for DRS  $c$ .  
 $P^c \subseteq P, c \in C$ .
- $Q$  Set of all working routes.
- $Q^r$  Set of eligible working routes belonging to demand relation  $r$ .  
 $Q^r \subseteq Q, r \in D$ .

**Parameters:**

- $d^r$  This parameter contains the number of unit demands for relation  $r$ .  
 $r \in D$ .
- $x^{p, q}$  Equal to 1 if the working route  $q$  is in an on-cycle relationship with cycle  $p$ , equal to 2 if the working route  $q$  is a straddling relationship with cycle  $p$ , and is equal to 0 otherwise.  $p \in P, q \in Q$ .

$\pi^{p, i}$  Equal to 1 if cycle  $p$  crosses span  $i$  and is 0 otherwise.  $p \in P, i \in S$ .

$\zeta^{r, q, i}$  Equal to 1 if route  $q$ , belonging to demand relation  $r$ , crosses span  $i$  and is 0 otherwise.  $r \in D, q \in Q^r, i \in S$ .

$c^i$  This parameter corresponds to the cost per unit of channel capacity on span  $i$ .  $i \in S$ .

### Variables:

$s^i$  Integer variable corresponding to the total spare capacity placed on span  $i$ .  $s^i \geq 0, i \in S$ .

$w^i$  Integer variable corresponding to the total working capacity placed on span  $i$ .  $w^i \geq 0, i \in S$ .

$n^{p, q, c}$  Integer variable corresponding to the number of unit capacity copies of cycle  $p$  used to protect working paths on route  $q$  as part of DRS  $c$ .  
 $n^{p, q, c} \geq 0, p \in P^c, q \in Q, c \in C$ .

$n^{p, c}$  Integer variable corresponding to the number of unit capacity copies of cycle  $p$  used to protect all instances of DRS  $c$ .  
 $n^{p, c} \geq 0, p \in P^c, c \in C$ .

$n^p$  Integer variable corresponding to the total number of unit capacity copies of cycle  $p$  used.  $n^p \geq 0, p \in P$ .

$g^{r,q}$ 

Integer variable corresponding to the number of working capacity units on route  $q$  to satisfy the demand relation  $r$ .

$$g^{r,q} \geq 0, r \in D, q \in Q^r.$$

### FIPP DRS JCP ILP:

$$\text{Minimize: } \sum_{i \in S} c^i \cdot (s^i + w^i) \quad (5.1)$$

### Constraints:

$$\sum_{q \in Q^r} g^{r,q} \geq d^r \quad \forall r \in D \quad (5.2)$$

$$\sum_{r \in D} \left( \sum_{q \in Q^r} \zeta^{r,q,i} \cdot g^{r,q} \right) \leq w^i \quad \forall i \in S \quad (5.3)$$

$$\sum_{c \in C^q} \left( \sum_{p \in P^c} x^{p,q} \cdot n^{p,q,c} \right) \geq g^{r,q} \quad \forall r \in D, \forall q \in Q^r \quad (5.4)$$

$$n^{p,c} \geq n^{p,q,c} \quad \forall r \in D, \forall q \in Q^r, \forall c \in C^q, \forall p \in P^c \quad (5.5)$$

$$n^p = \sum_{c \in C} n^{p,c} \quad \forall p \in P \quad (5.6)$$

$$s^i \geq \sum_{p \in P} n^p \cdot \pi^{p,i} \quad \forall i \in S \quad (5.7)$$

The objective function (5.1) minimizes the total cost of spare and working capacity in the network. Constraint (5.2) ensures that all of the unit demands for every demand relation is routed along one or more eligible working routes belonging to that demand relation. Constraint (5.3) makes sure that the working route decisions

made for every demand are adequately capacitated. Constraint (5.4) places enough FIPP  $p$ -cycles to protect the working routes in the DRSs selected to be part of the solution. Constraint (5.5) updates the variable that keeps track of exactly which cycles protect which DRSs and their quantities. Based on the prior, constraint (5.6) calculates the total number of copies of each cycle. Finally constraint (5.7) makes sure that enough spare capacity is allocated for the FIPP  $p$ -cycles selected.

This model has the property of demand splitting, meaning that the paths carrying unit demands belonging to the same demand bundle do not have to be routed over the same spans. Any combination of eligible routes belonging to a single node pair may be used as long as the sum of the demand units over these routes satisfies the demand requirement for that node pair. This model can further be extended to include capacity modularity and economy of scale effects using the methods outlined in [72]. Briefly, this refers to allowing the model to select between several transmission rate modules, such as OC-48 and OC-192, with the knowledge, for example, that while the latter provides four times as much capacity as the former, it can be installed for only twice the cost.

## 5.4 Experimental Setup

The FIPP DRS JCP ILP model was implemented in AMPL 9.0 and solved using the CPLEX 9.0 MIP solver on a quad-processor Sun V480 SPARC server with 16GB of RAM. All the solutions are based on complete CPLEX terminations with MIPGAP of 0.01. None of the FIPP  $p$ -cycle solutions generated took longer than 1.5 hours to obtain. The test-case network is COST239 from [11]. A total of 15 active demand relations were used and each of these relations exchanged 4 units of demand. The active demand relations were selected randomly, representing a possible network state for this topology. Further details regarding the demand set and the COST239 network can be found in the Appendix (Section 10.1). Span costs are equal to the geo-

graphic distance between the nodes as provided in the initial published data for this network [11].

Except where deliberately varied in following results, for every relation exchanging non-zero demand, 5 eligible shortest working route candidates were generated. The value of this parameter was set as high as possible while still keeping the problem to a reasonable size to give the solver a large number of options. Each relation is represented by its shortest route candidate, which was used as the *seed* route for the formation of a total of 10 DRSs. This means that the shortest route on each non-zero demand relation was guaranteed to appear in at least 10 DRSs. The eligible routes that were not the shortest were introduced into the DRSs using the method described in Section 5.3. Also inserted in the set of all DRS candidates was a single DRS for each demand that contained only its single shortest route. This ultimately ensures feasibility of a solution even if it means falling back to simple 1+1 APS (which single route DRSs protected by a unit cycle can be converted to) as the solution for a given demand pair. The intent was, specifically, to enable this possibly if warranted in the global solution. (For individually very strong ‘forcers’ in a survivable network design problem in general, a dedicated 1+1 APS treatment can sometimes be the best solution. The forcer concept is covered in [73].) For each DRS, the default was to provide 5 shortest candidate cycles that pass through the set of end-nodes of the working routes in the DRS. The maximum number of routes that any one DRS could contain was also set to 10. These default values were used in every experiment unless stated otherwise.

The values for the parameters above, excluding the number of eligible working routes per relation, were set based on a preliminary set of tests designed to identify a ‘sweet spot’ for each parameter where a compromise was met between the efficiencies of the solutions obtained and the size of the problem. These initial tests varied the parameters one by one, and the results were studied in order to identify where the

solutions obtained from subsequent increases in the parameter in question did not significantly improve as higher values of the parameter were used.

For comparative reference, corresponding optimal span-protecting  $p$ -cycle JCP solutions were generated using the model in [69]. Those solutions are based on a full CPLEX termination with a MIPGAP of 0.0001 with all possible cycles and five eligible routes per relation used as input to the problem. Optimal JCP SBPP solutions were also obtained using the model in [74]. In the latter solutions, a maximum of 10 eligible disjoint protection paths and 5 eligible routes per relation were used as input to the problem and all solutions are based of full CPLEX terminations with a MIPGAP of 0.0001.

## 5.5 Results and Discussion

### 5.5.1 Basic Results

With the parameter values stated above, we were able to generate the basic results shown in Table 5.1. This table contains the total network costs from solutions obtained using FIPP DRS JCP method as well as the  $p$ -cycle JCP, SBPP JCP (for all three JCP models, 5 eligible shortest working route options were provided to the solver) and FIPP DRS (SCP) solutions for comparative reference. It was found that when the FIPP DRS JCP model was run using exactly 5 shortest eligible working route options, the solutions generated were very poor (more costly than SCP solutions) because all the DRSs generated contained a mix of up to and including the 5th shortest working route sizes and these routes were randomly placed into the DRSs using the algorithm in Section 5.3.

It turned out (as will be shown later) that the DRSs generated used more working capacity than was necessary and that it was possible to achieve better results using shorter working route options. Additionally, the longer routes sometimes pre-

vented the DRSs from being maximally filled, thus resulting in fewer DRSs containing the maximum number of working routes. This, in turn, gave the solver fewer options and forced additional cycles to be added to the solution to protect the working routes that would have otherwise been used as part of more efficiently packed DRSs, given fewer routing options. In order to make sure that a good representative FIPP JCP solution was obtained, 5 sets of DRSs (all of them individually adhering to the parameters set in Section 5.4) were given to the solver.

Of these 5, each set had an increasing number of eligible working route options so that the first set provided only DRSs using the shortest working routes, the second provided DRSs using the first and second shortest working route options, the third using the first, second and third shortest working route options and so on. This basically insured that the solver would be able to select DRSs containing the 5th shortest routes only if it is optimal to do so. The value obtained using this set up is presented in Table 5.1 in the FIPP DRS JCP column.

**TABLE 5.1. Initial results for FIPP,  $p$ -cycle and SBPP JCP methods using 5 eligible working route options per demand. The table also contains the results for the FIPP SCP solution.**

	FIPP DRS JCP	FIPP DRS SCP	$p$ -Cycle JCP	SBPP JCP
<b>Total Network Cost</b>	65625	74145	62780	64920

The initial results (according to Table 5.1) indicate that relative to the SCP case, the FIPP DRS JCP solution was able to reduce the total cost of the network by 13%. The lower cost of the JCP solution relative to the SCP solution was expected based on the arguments from Section 5.2. Recall that, similarly to the FIPP DRS SCP method, the FIPP DRS JCP solution obtained by the solver is simply the combination of the pre-generated DRSs and cycles that yielded the lowest network cost while providing 100% single span failure restorability for every non-zero demand relation. Despite this, the FIPP JCP solution was only 4% more costly than the corresponding optimal  $p$ -cycle solution and just over 1% more costly than the corresponding optimal

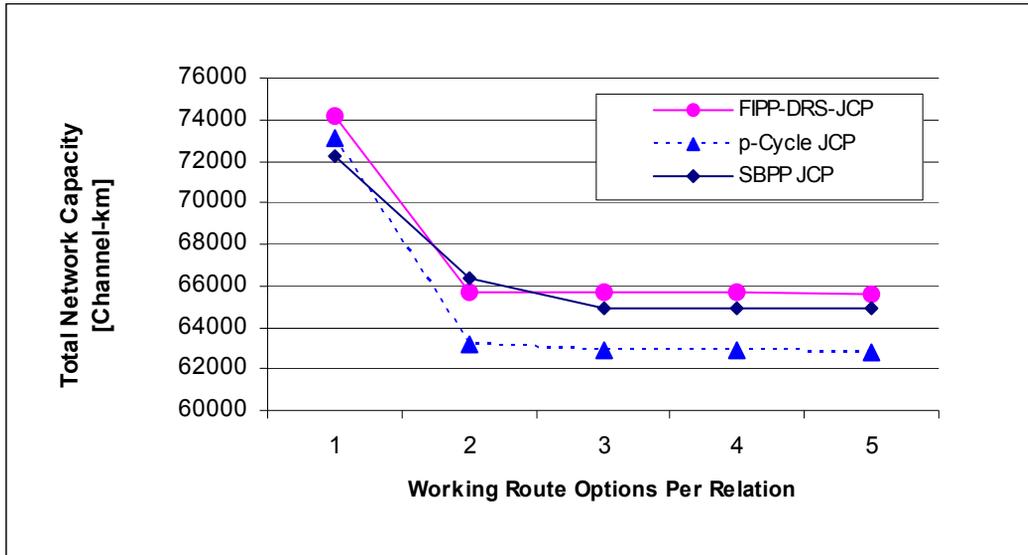
SBPP solution. This means that the FIPP DRS JCP method, despite not necessarily finding the absolute optimal FIPP  $p$ -cycle solution, is capable of generating high quality network designs that are comparable to solutions obtained for SBPP and span-protecting  $p$ -cycle architectures; both of which are well known for their spare capacity efficiency.

### **5.5.2 Varying the Number of Eligible Working Routes per Relation with DRS propagation**

This section expands the results presented in Section 5.5.1 by analyzing the total cost of the network as the number of eligible route options is varied. In this experiment the set of eligible routes per node pair is varied between containing only the shortest route, and the set of the first five successively shortest routes. In other words this section is a stage-by-stage breakdown of the experiment in Section 5.5.1. The single shortest route case represents the non-joint SCP case and, as larger sets of eligible routes are represented, the extent of the possibilities for joint coordination of working routes with protection structures is (in principal) continually increased. The effect of the increase in these possibilities is the central effect to be studied in this section.

The experiment was run in five iterations, starting with only the shortest working route per demand (the SCP case) and ending with five shortest routes per demand. In addition to the DRSs generated for each iteration, all the DRSs from the prior iterations were included as input to the solver. For example all DRSs from the one and two eligible route cases were carried forward to be merged with the set of DRSs of the third iteration (containing up to and including the 3rd shortest eligible route) and so on. This was done to ensure that the lowest cost solution found at any iteration would propagate to the end of the experiment and would only be overwritten if adding extra working routes to the problem did indeed provide a reduction in cost, relative to the best solution found thus far. This is the same way that the results

in Section 5.5.1 were obtained, except that only the end results were presented before, where as now each individual increase in the number of eligible working routes can be examined.



**FIGURE 5.3. Network cost vs. number of eligible working routes per demand with DRS history propagation.**

The results found in Figure 5.3 illustrate that as the number of eligible routes per demand is increased from one to two, there is a significant decrease (13%) in the total network cost of the *p*-cycle FIPP design. Beyond two eligible working routes per demand, there is no significant change in the cost of the FIPP DRS JCP solutions. This means that the additional working route options organized into DRSs using the algorithm from Section 5.3 did not provide cost savings relative to the solution obtained using just two eligible working routes. Optimal JCP *p*-cycle solutions, also included in Figure 5.3, exhibit a similar trend. The joint solutions obtained using the FIPP DRS JCP model were at most only 4% more costly than the optimal *p*-cycle solutions. For further comparison, optimal SBPP solutions were also included as shown by the connected diamonds in Figure 5.3.

Upon closer examination of Figure 5.3, we can see that in the SCP case, the first set of data points where only one working route option was provided, SBPP yielded the least capacity solution. This was closely followed by span-protecting  $p$ -cycles and then FIPP  $p$ -cycles. The second set of data points, where both the first and second shortest routes per demand were allowed as input to the problem, shows that  $p$ -cycles yielded the best solution, followed by FIPP  $p$ -cycles and then SBPP. The reason that SBPP performed poorer in this case was because SBPP does not allow for demand splitting between various alternate working and backup paths for its demands. Meaning that given several routing alternatives for a particular demand, SBPP chooses only one for all the demand exchanged by that node pair. This is different from span-protecting and FIPP  $p$ -cycles, where the solver is allowed to split demand bundles over several routing alternatives. This property gives the  $p$ -cycle approaches much more flexibility, and explains why joint SBPP may sometimes perform poorer compared to joint regular and FIPP  $p$ -cycle designs.

An interpretation of this above result can be offered. It is to say that in this network 3rd shortest routes tend to be so much longer than the shortest routes that it is unlikely to obtain correspondingly greater savings in spare capacity through their use when 2nd shortest routes are present. Figure 5.4 illustrates the relative lengths of the eligible working route options compared to the shortest routes. Many of the 2nd shortest routes are within 25% of the shortest route, while the 3rd and subsequent routing options fall into the range of being 50%+ longer. The COST 239 network used is a fairly high-degree network (4.7). This effect should be even greater in a lower degree network, leaving us with a welcome and practical guideline that most benefits of a joint design can be obtained by problem formulation so as to represent only the shortest and second shortest routes.

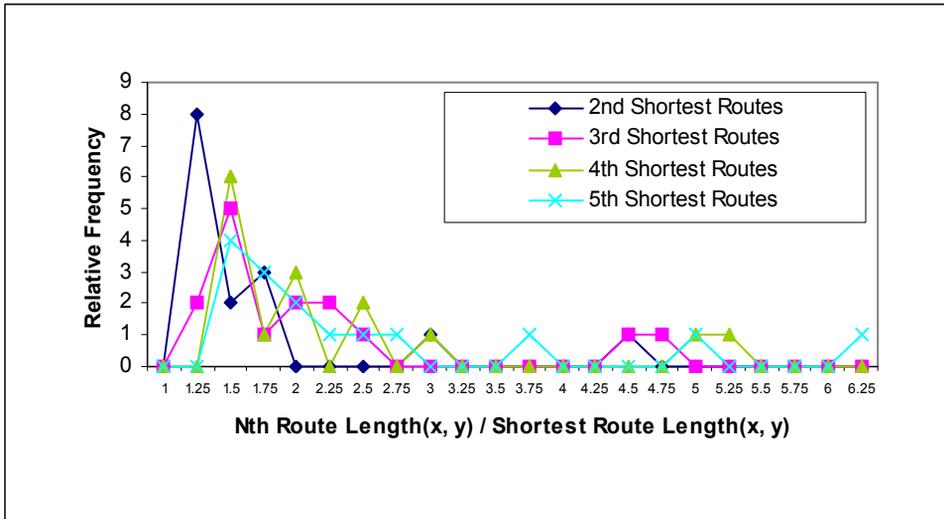


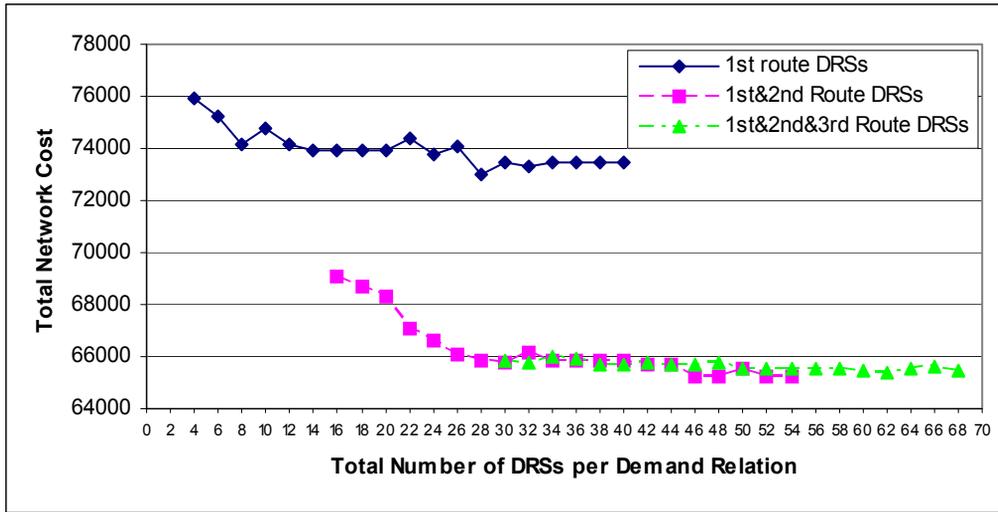
FIGURE 5.4. Lengths of eligible working route options relative to the shortest routes.  $(x, y)$  refers to the node pair that the route connects.

### 5.5.3 Varying the Type and Number of DRSs per Demand

While the previous experiment does demonstrate the utility of joint design, it does not actually give very much information about solutions obtained using three or more eligible working route choices per demand in terms of the number of DRSs given to the problem and the sets of eligible route options they contain. Using previous solutions, it is difficult to determine exactly how much jointness contributes to the initial cost reduction and how much of that improvement is due to the increase in the number of DRSs per iteration. The following experiment was set up so as to diminish the effect of increasing the number of DRSs and to demonstrate the effect of adding additional eligible working routes on overall network cost. Three types of DRSs were considered, the main difference between them are the types of working routes they contain. Type 1 DRSs contain only the shortest working routes. Type 2 DRSs contain the first and second working route options. Type 3 DRSs contain the first, second and third shortest routes. These are the same types of DRSs that were

generated in the first, second and third iterations of the previous experiment (Section 5.5.2).

This experiment was split up into three trials. In the first trial, corresponding to the diamond data points in Figure 5.5, the number of DRSs per demand was varied from 4 to 40. Recall that “the number of DRSs per demand” refers to the number of unique DRSs in the DRS set containing at least one eligible route of the demand in question. All the DRSs in the first trial were of Type 1, which corresponds to the SCP case where only the shortest working routes per node pair were used (recall that each DRS represents a combination of working routes that can share protection under the same cycle). In the second trial, represented by the square data points in Figure 5.5, the number of Type 1 DRSs was set to 14 and the number of Type 2 DRSs was varied from 2 to 40. The main idea behind setting the number of Type 1 DRSs to 14 is that this is the point at which the graph for trial 1 begins to plateau first. Selecting this point minimizes the impact of additional DRSs on network cost while keeping the size of the problem as small as possible (for this experiment we are interested in the impact of additional working routes, not additional DRSs). In the third trial, represented by the triangle data points in Figure 5.5, the number of Type 1 and 2 DRSs was set to 14 each, while the number of Type 3 DRSs was varied from 2 to 40. Once again, the reason that the number of Type 1 and Type 2 DRSs were set to 14 was to minimize the impact of additional DRSs to the problem while keeping the problem size as small as possible.



**FIGURE 5.5. Total network cost versus number of DRSs of various types.**

The results from the first trial (diamond data points) indicate a decrease in cost as the number of DRSs per demand is increased. From a standpoint of choosing working routes to share the same protection structure, this is the result of the solver being able to generate better solutions when given more choices. There is a significant difference between the data obtained from the first and second trial (square data points) at the point where the total number of DRSs per demand is 16. There is no significant change in network cost between using 14 and 16 Type 1 DRSs per demand as indicated by data from trial 1. However, just adding two Type 2 DRSs to 14 Type 1 DRSs per demand allows for a 7% reduction in total network cost. It can be further seen that by giving the solver 14 Type 1 DRSs and 14 Type 2 DRSs per demand, as opposed to using 28 Type 1 DRSs per demand, a 10% reduction of cost is possible. There is no significant change to network cost when Type 3 DRSs are added to the mix as indicated by the data from the third trial (triangle data points). This confirms the data obtained in Figure 5.3 where there was no significant cost reduction between using two and three working route options per demand.

## 5.5.4 Varying the Eligible Route Set Inclusion Threshold

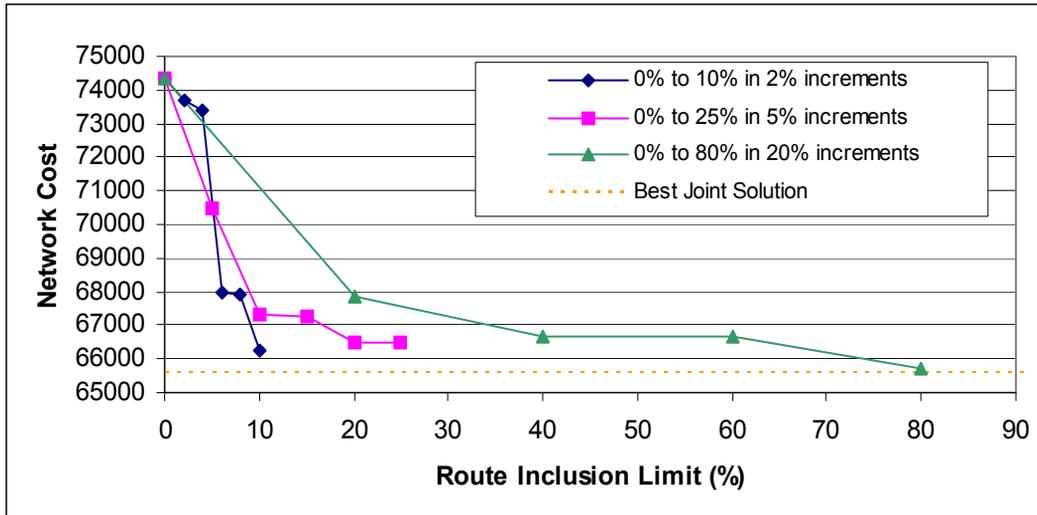
The previous experiments involved varying the number of working route options represented per demand. This metric only reflects the route's rank (its place in a sorted list of a demand's set of route options) and does not reflect the route's actual geographic length. This means that by simply picking the first  $n$  shortest routes per demand we may be picking candidate working routes that are many times longer than the shortest route of that demand. Recall that (at least in the case of [27]) it was found that working routes selected in joint solutions tend to be only a few percent longer than the shortest route. This means using longer working routes when generating DRSs may actually have no real chance of helping the quality of the solution. On the other hand, limiting routes by their rank instead of length may actually exclude beneficial working routes options from being selected at all. This may occur where a particular node pair has more than  $n$  working routes that are just a few percent longer than the shortest route. Clearly, selecting just  $n$  working routes may exclude good route candidates in this case.

Accordingly, the following experiment uses relative length of the working routes to the shortest routes as the criteria that determines which working route options are included in the DRS generation process. The parameter that controls this is the route inclusion threshold. For example, if the route inclusion threshold is set to 0% then only routes that are no longer than the shortest working routes are used to generate the DRSs. If the threshold is set to 100% then all working route options that are up to and including twice the length of the shortest working routes are used to generate the DRSs.

The experiment was run in three trials. The route inclusion threshold was varied from 0% to 10% in 2% increments per iteration, 0% to 25% in 5% increments per iteration and 0% to 80% in 20% increments per iteration for the first, second and third trials, respectively. Again, the DRSs generated in earlier iterations were carried

forward to the later ones. This is similar to the DRS propagation used in the experiment outlined in Section 5.5.1. Its purpose was to make sure that the least cost solution found at any iteration would carry forward to the end of the trial.

The three trials are illustrated in Figure 5.6. Correspondingly the number of working routes per demand that fall under the threshold of each trial is shown in Table 5.2. The main inference from the results is that the most significant cost reduction comes from including working routes that are as little as 10% longer than the shortest working route per demand. At the 10% threshold it was possible to get a solution that was only 3% worse than the best joint FIPP solution found so far in Section 5.5.1, where the working routes were not selected by length. This is a significant result because, while the experiment in Section 5.5.1 had to consider a total of 30 working routes (15 demands with 2 working route options each) the experiment using the length threshold only had to consider 19 according to the first trial in Table 5.2. However, the strictly best solution in Figure 5.6 appears only after the route inclusion threshold is set to 80% and is only 0.14% worse than the best joint FIPP solution (which is well within the MIPGAP of 1% and so not significantly different at all). For this result, according to the third trial in Table 5.2, a total of 87 working route options had to be considered when using the route inclusion threshold method. This is almost three times that of the best solution from Section 5.5.1 where, as mentioned before, only 30 working route options had to be considered to achieve the same result.



**FIGURE 5.6. Network cost vs. eligible route inclusion threshold with DRS history propagation. The best joint solution refers to the FIPP DRS JCP data point corresponding to 2 eligible routes per demand in Figure 5.3.**

Note that at the route inclusion threshold of 10% (first trial in Table 5.2), only four node pairs actually have second alternate working routes, while the rest only have one. Given to the solver as input, these few alternate routes reduce the cost of the network by 10%, relative to the SCP solution. As the threshold is increased to 25% (second trial in Table 5.2), only five demands are left without alternate routes. By the time the threshold is set to 80% (third trial in Table 5.2), two demands remain without alternate route choices while some demands have as many as ten. This trend, where some demands gain a large number of alternatives while other demands have no alternatives at all becomes much more apparent when the threshold is increased past 60%. For example, Table 5.2 shows that there is a still a demand (N1-N2) without alternative routing even at the 200% threshold (three times the length of the shortest route), while other demands have tens, some over a hundred alternate routes available. When the number of eligible routes per demand gets too large, the performance of the route inclusion threshold method begins to decline. When this occurs, some working routes will not be represented in any DRSs at all, and there is no guarantee that the

ones that are represented are strong enough to be considered as alternates to the shortest route.

**TABLE 5.2. Number of working route choices per demand that fall under the threshold of the first, second and third trials. The number of working route choices per demand for a threshold of 200% is also given.**

	First Trial						Second Trial						Third Trial					# of Working Routes per Demand for Threshold of 200%
	0%	2%	4%	6%	8%	10%	0%	5%	10%	15%	20%	25%	0%	20%	40%	60%	80%	
N0-N1	1	1	1	1	1	1	1	1	1	2	2	2	1	2	3	5	5	27
N0-N2	1	1	1	1	1	1	1	1	1	1	1	2	1	1	2	3	3	11
N0-N7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	5
N1-N2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
N1-N7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	16
N1-N8	1	1	1	2	2	2	1	2	2	2	2	2	1	2	4	6	8	48
N2-N5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	6	7	61
N2-N9	1	1	1	1	1	2	1	1	2	2	2	3	1	2	8	9	14	76
N3-N7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	25
N3-N8	1	1	1	1	1	1	1	1	1	1	2	2	1	2	4	7	12	100
N4-N7	1	1	2	2	2	2	1	2	2	2	3	3	1	3	3	4	12	123
N5-N6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	11
N6-N7	1	1	1	1	1	1	1	1	1	2	2	2	1	2	2	2	2	4
N6-N8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2
N7-N10	1	1	2	2	2	2	1	2	2	2	2	2	1	2	6	9	14	176
<b>Total</b>	15	15	17	18	18	19	15	18	19	21	23	25	15	23	41	59	87	686

As a further diagnosis to gain insight, Table 5.3 contains the distribution of working flow in the solutions for each trial of the route inclusion threshold experiment. It shows how the total number of demand units was distributed among the various routing alternatives for different threshold values. For example, the solutions to the first trial in Table 5.3 used at most only the first and second working routes, which is all that was available to them. When the threshold was 0%, all 60 units of demand were routed over the shortest routes possible. However by the end of the trial where the threshold was set to 10%, 9 units of demand were deviated to their second shortest routes leaving 51 units using their shortest route options. The end result of the second trial in Table 5.3 deviated 1 unit of demand to its third shortest routing option, 10 units to the second shortest option and, leaving the remaining 49 units routed over

the shortest routes. The third trial in Table 5.3 was given as many as nine routing alternatives for some demands according to the table and used mostly the first three shortest routes in the solutions up to and including the 60% threshold. The fifth shortest route was used for 2 demand units at the 80% threshold. This shows that the solver is actually quite able to select routes for working paths that are much longer than the shortest paths when it is globally optimal to do so. These are cases where working capacity may be added as long as there is a corresponding reduction in spare capacity so that the total cost of the design is improved. In other words the placement of the longer working routes allowed the solver to select a more efficient arrangement of spare capacity allowing for a significant reduction in total cost. While the solver chose some significantly longer working routes in the high threshold limit trials, the most significant gains were still the result of deviating relatively little from the shortest working paths.

**TABLE 5.3. Working flow distribution for the route inclusion threshold for trial 1, 2 and 3. Each column corresponds to an iteration where a particular threshold was set. For each iteration, the types of working routes used as well as the number of demand units routed among them is shown.**

<b>Trial 1</b>	<b>0%</b>	<b>2%</b>	<b>4%</b>	<b>6%</b>	<b>8%</b>	<b>10%</b>
<b>Shortest</b>	60	60	58	54	54	51
<b>2nd Shortest</b>			2	6	6	9
<b>3rd Shortest</b>						
<b>Trial2</b>	<b>0%</b>	<b>5%</b>	<b>10%</b>	<b>15%</b>	<b>20%</b>	<b>25%</b>
<b>Shortest</b>	60	54	52	51	50	49
<b>2nd Shortest</b>		6	8	9	9	10
<b>3rd Shortest</b>					1	1
<b>Trial 3</b>	<b>0%</b>	<b>20%</b>	<b>40%</b>	<b>60%</b>	<b>80%</b>	
<b>Shortest</b>	60	52	48	48	45	
<b>2nd Shortest</b>		8	9	9	11	
<b>3rd Shortest</b>			3	3	2	
<b>5th Shortest</b>					2	

### 5.5.5 Visual Inspection of Joint FIPP $p$ -cycle Solutions

This section contains illustrations of the actual cycles and the DRSs that make up the SCP and the best JCP solution from Section 5.5.1. The illustrations were added in order to provide a validation of the functional correctness of the designs and also to allow for inspection and comparison of the two solutions. Figure 5.7 and Figure 5.8 illustrate the SCP and the JCP solutions respectively. In these figures the cycles are represented by the thick connected lines, while the working paths are represented by the solid and dashed arrowed lines, corresponding to the first and second shortest routes respectively.

The SCP solution where only the single shortest working routes per demand were given to the solver (shown in Figure 5.7) contained a total of 9 distinct cycles. Of these cycles, 6 protected DRSs which contained 7 working routes or more, while the remaining 3 cycles protected DRSs that contained only a single route. Recall that these single route DRSs were added to the problem in order to give the solver the option of efficiently protecting single demands by avoiding placing large cycles for these demands' needs only.

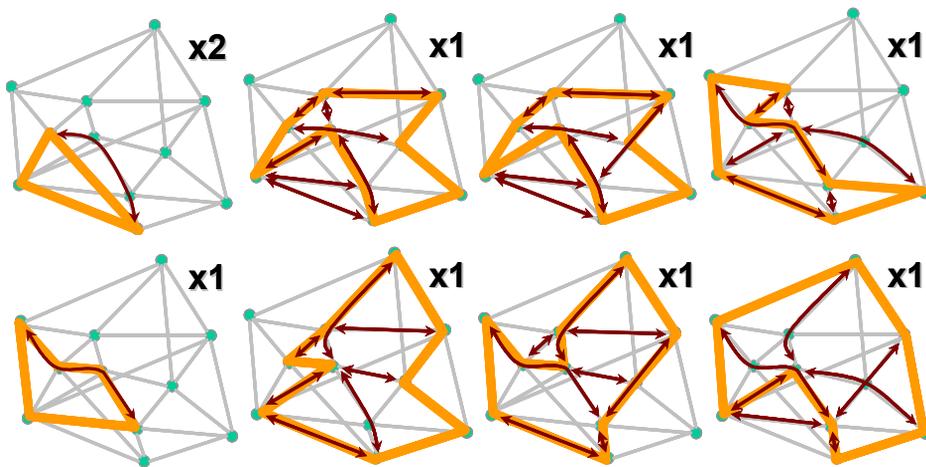


FIGURE 5.7. DRS based SCP FIPP  $p$ -cycle solution (corresponding to the first data point of the FIPP DRS JCP solutions in Figure 5.3).

The lowest cost joint solution from Section 5.5.1, where the solver was given the option to select between the first and the fifth shortest routes per node pair, is illustrated in Figure 5.8. Recall that since the working route options above the 2nd shortest did not provide any significant cost savings, the lowest cost solution contains only the first and second shortest routes. Only 5 cycles were used in this solution and no single route DRSs were protected at all. Of the 60 demand units to be protected, 13 were deviated away from their shortest routes. This allowed the solver to select better disjoint route combinations and in turn allowed the solver to select fewer cycles to completely protect all the demands against single failures. Table 5.4 summarizes the capacity costs of the two solutions as well as the number of cycles used. The joint solution allowed for deviation from the shortest working paths and this resulted in higher working cost, relative to the SCP solution. However, this slight increase in working cost was offset by the reduction in the number of cycles needed and thus the spare capacity cost. This, in turn, contributed to a 13% reduction in total network cost.

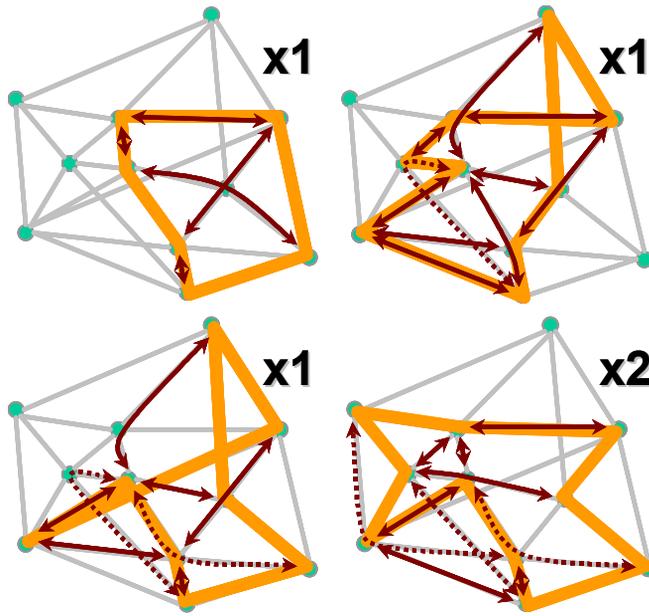


FIGURE 5.8. DRS based JCP FIPP  $p$ -cycle solution (corresponding to the second data point of the FIPP DRS JCP solutions in Figure 5.3).

TABLE 5.4. Design costs and the number of cycles used in sample SCP and JCP solutions.

	Cost of Working	Cost of Spare	Total Cost	Number of Cycles
<b>SCP Solution</b>	42820	31325	74145	9
<b>Joint Solution</b>	44840	20855	65695	5

In the specific case of FIPP  $p$ -cycles, one of the biggest contributors to improved network cost in joint optimization is the fact that it allows for a single cycle to protect more than two units of working capacity per demand. This happens because joint design benefits from topological diversification, where several working routes for the same demand can share the same protection structure (cycle) because they cannot be simultaneously affected by a single span failure, provided they are disjoint from one another. In other words, SCP solutions limited each instance of a unit FIPP  $p$ -cycle to protecting a maximum of two units of working capacity in the straddling case. Joint solutions allow cycles to protect as many routes between a single node

pair as there are span disjoint working routes between that node pair. The number of working capacity units that can be protected on those routes correspond to the relationship the route has with the cycle assigned to protect it. This allows for FIPP  $p$ -cycles to effectively deal with highly loaded demands without having to place many instances of a single cycle for the benefit of that demand. Furthermore, jointness allows for the solver to gain benefit from additional straddling relationships that would otherwise not be possible in SCP. Consider being able to deviate from the shortest route which is in an on-cycle relationship to a route that is nearly as short, but is a straddler. The additional unit of working capacity protected by the already placed cycle comes at nearly no additional cost. Also, if that unit of working capacity required a dedicated cycle to be placed for it in the SCP case, that cycle could be removed from the network entirely. This effect, as well as the ability of the solver to pick more efficient DRS/cycle combinations is why there are no cycles dedicated to single routes in the joint solution compared to the SCP solution.

## 5.6 Conclusion

A design algorithm for producing jointly optimized FIPP  $p$ -cycle networks was proposed, implemented, and used to study the benefits and other effects of jointly coordinating working route choices with FIPP  $p$ -cycle placements. The resulting networks are, by design, 100% restorable against any single span failure at costs that are significantly lower than non-joint designs. On the COST 239 test-case network, results showed a 13% improvement vis-à-vis non-joint designs just from allowing a choice between the two shortest eligible route candidates. This result was at most only 4% more costly than optimal  $p$ -cycle solutions, which is an excellent result given the fact that the FIPP  $p$ -cycle solution employs only fully pre-cross-connected end-to-end path protection structures. Through experimentation and testing of how the benefits of jointly optimized design work in the specific case of the FIPP  $p$ -cycle architecture, we also produced a number of related insights on how to simplify instances of the

joint FIPP design problem. Specifically, we offer the following in summary as guidelines and observations in addressing the joint design problem for FIPP  $p$ -cycles:

- Significant reduction in network cost, relative to the SCP solution, is obtainable by admitting consideration of alternative working routes that are only relatively slightly longer than the strictly shortest routes. Here admission of working routes within 10% of the length of the shortest was effective. Additional gains are possible but the number of additional working routes that need to be considered becomes very large.
- Quite good solutions seem obtainable by considering only the first and second shortest eligible working route options under joint optimization. This means it is not necessary to provide more than two eligible working routes per demand as input to the solver as, in our experience, doing so only unnecessarily increases the size of the problem instance, with little further contribution to solution quality.
- The main benefit of jointness in FIPP  $p$ -cycle design comes from the ability of the individual unit cycles to protect more than a maximum of two units of capacity per node pair, in contrast to the SCP case. Additionally, deviation from the shortest working route to slightly longer working routes allows the solver to capitalize on straddling relationships that were not available under the corresponding SCP problem instance.



## 6.1 Introduction

The following chapter is based on research done as part of a three year (November 2005 - November 2008) collaboration between TRILabs and Nokia Siemens Networks under the name of High-Availability Network Architectures (HAVANA). The study, spanning a total of three years, primarily focused on investigations of pre-connected network survivability architectures. The main participants in this study include the author (FIPP  $p$ -cycles), Dr. Aden Grue (PXTs), Brian Forst (DSP and APS), Diane Onguetou (span-protecting  $p$ -Cycles), as well as Dr. Wayne D. Grover, Dr. Matthieu Clouqueur and Dr. Dominic Schupke. This chapter summarizes the first two years of the study in the context of FIPP  $p$ -cycles. For these two years additional contributions were made by Dr. John Doucette as well as Dr. Adil Kodian. The overall goal of this body of work was to investigate pre-connected architectures in the context of transparent optical networking. Select results from this part of the study were published in [64]. Furthermore the details regarding the approaches taken to achieve solutions for every pre-connected architecture under study can be found summarized in [65], [66].

In this chapter, the underlying theme of pre-connection is introduced first. This is followed by the generation of the initial minimum cost and capacity single span failure restorable designs (Section 6.3) which act as the initial stages for subsequent analysis of select aspects of optical transport networking. These initial solutions are generated for every architecture under study and the results are used to get a feel for how the architectures scale with respect to one another. Once this is done, as part of the first study these initial designs are subjected to dual-failure scenarios in Section 6.4. In this section, strategies are developed to help FIPP  $p$ -cycle single span failure designs cope with dual-failures scenarios. Section 6.5 and Section 6.5.2 deal with minimum wavelength assignment of minimum capacity solutions under specific

hardware constraints. New minimum capacity solutions based on the FIPP DRS JCP method are introduced in Section 6.5.3 in order to meet the maximum number of wavelengths hardware constraint. Transparent reach analysis follows in Section 6.6 where several of the designs generated thus far are examined from the perspective of optical networking where fiber attenuation and insertion losses are taken into account. Node failure analysis of minimum capacity solutions is provided in Section 6.7 and is followed by a section on 100% single node failure restorable FIPP  $p$ -cycle design in Section 6.8.

## 6.2 Background

In the context of survivable transport networking, the property of full-pre-connection carries with it many benefits. Pre-connection, in this context, refers to the state that the spare capacity is in prior to the onset of failure. Specifically, pre-connection refers to the organization of spare capacity into complete protection structures that provide protection paths to the working paths in the network. If these protection paths are fully pre-connected, it means that they are in guaranteed full working condition and have been engineered, tuned and tested. If the protection paths are kept in this state, they can be switched to in the case of a failure without any degradation in signal quality aside from the time lost due to failure identification and the actual switching action. This type of pre-connection is especially important in optical networking, where routing and restoration of entire lightpaths is considered. The assurance of optical transmission integrity may not be assured if the protection path needs to be cross-connected and assembled dynamically after failure. A considerable amount of time would be lost as the adaptive protection mechanisms concatenate spare capacity resources into end-to-end paths. It is unrealistic to expect that this concatenation would result in high quality, low BER ( $< 10^{-12}$ ) transmission right away. Fiber attenuation, dispersion as well as amplifier non-linearities and noise would all have to be

precisely calibrated for, before high quality transmission may be achieved. Full pre-connection allows for these effects to be accounted for before failure.

In the fall of 2005, the Network Systems group at TRILabs, Edmonton was commissioned by Siemens Optical Networking & Transmission Group to investigate the aspects of various pre-connected architectures in the context of optical transport networking. Specifically, the goal was to examine the relative strengths and weaknesses of each architecture and how these would compare to one another if they were implemented in real optical networks.

### 6.3 Minimum Capacity Designs

The initial step in this project was to generate minimum capacity and minimum cost 100% single span failure recoverable designs for each architecture. This was done to make an initial assessment on how these architectures compare to one another capacity and cost wise as well as to generate basic reference solutions intended for use in subsequent case studies. In an attempt to provide a common ground for comparative studies, most architectures under study used SCP methods to generate the initial solutions. This meant that the working routing for all the architectures was intended to be the same, or at very least close to the same, given that some architectures such as DSP and FIPP  $p$ -cycles had to deviate from this slightly because of architecture specific peculiarities. In general, it should be noted that all working capacity was routed with the overall goal of minimizing it.

For the FIPP  $p$ -cycle architecture, the FIPP disjoint route set (DRS) method [40] was used to generate designs for the test network. The DRS method can be found in detail in Section 4.6. Briefly, the DRS method is an ILP method where a large number of DRS/cycle configurations are generated and are passed to the solver, which selects the lowest cost combination of these configurations such that all demand requests are fully satisfied and the design is 100% single span failure restor-

able. The DRSs/cycle configurations are generated in two steps: The first step entails the generation of DRSs, where mutually span disjoint working routes are placed into route sets (DRSs) and the second step entails the generation of a number of lowest cost cycles for each DRS, eligible to protect it. It should be noted that, the DRSs are generated with the goal of fairly providing enough protection options for every demand relation.

### **6.3.1 Equipment Capability Assumptions and the z-Case**

In the initial stages of this study, the exact details of the equipment costs were omitted and the main metrics used to compare the different architectures were total cost and capacity use. However, certain assumptions were made regarding the functionality of the equipment that these architectures may eventually be implemented on. These assumptions were based on the functionality of modern transport networking systems, but were also made general enough that the architectures under study could all be implemented using the same equipment. This made it much easier for comparisons to be made between architectures and it allowed each HAVANA project member to view their tasks in terms of the following question: “How well can your assigned architecture perform given specific equipment assumptions and the goal at hand?”

In this study, whenever cost is used as a metric, it is assumed that the cost of the equipment needed to establish transmission on a single span is proportional to the geographic length of that span. The task of generating minimum cost designs translate to minimizing the total geographic lengths covered by working paths and protection structures. This is a very commonly used metric in the area of transmission networking. Whenever capacity was is as a metric, it was assumed that the main contributor to cost is the nodal transmission equipment and that the cost of equipment is the same regardless of the length of the spans crossed by working and protection paths. In this case, the cost of fiber is assumed to be free and the task of minimizing cost translates

to that of minimizing the total number of hops used in the network, which corresponds to crossing fewer nodes and therefore needing less equipment installed overall.

The main assumption made about equipment functionality is that the equipment is only capable of simple pre-determined switching actions. This means that every working path is to have single pre-determined protection path that the end-nodes of the demand relation are to switch over to in case of a failure. For FIPP  $p$ -cycles, this assumption draws attention to a particular type of cycle/working path relationship: the  $z$ -case (Section 3.3.3). Recall that a  $z$ -case is a cycle/working path relationship where both of the possible protection paths provided by the cycle to the working path are not span disjoint from it. This means that two distinct span failures affecting the working path can also affect one of the protection paths. Since the protection paths are mutually span disjoint (and also mutually node disjoint, since simple cycles are used), the two protection paths can never be simultaneously affected by the same span failure. In the general case, this type of relationship is not a problem because whenever one of the protection paths and the working path are affected by failure, the second protection path remains intact and can be used to recover the failed working path. Note that this requires the end-nodes to be able to decide between two different protection paths and to be able to switch over to an appropriate one, depending on which span was affected by failure. However, since the equipment used in this study is assumed to be very simple and only capable of performing single pre-determined switching actions, the  $z$ -case becomes a problem when the predetermined backup path fails along with the working path. Ultimately this means that if the hardware cannot support the  $z$ -case relationship, any design that allows these relationships to exist will not be 100% single-failure restorable.

In order to avoid the  $z$ -case, the DRS method used to generate the minimum cost/capacity solutions was slightly modified. In the modified method, every DRS is inspected for  $z$ -cases before the problem is passed to the solver and the offending paths are flagged so that the solver does not include these relationships in the final solution.

This is done by setting the parameter  $x^{p,r}$  to 0 wherever there is a z-case between demand  $r$  and cycle  $p$ . It should be noted that using this type of preemptive z-case removal may result in solutions where some cycles appear to be slightly longer than necessary for the DRS they are assigned to protect. This happens when working paths are uniquely assigned to cycles in the case of overprotection as will be described in detail in the next subsection.

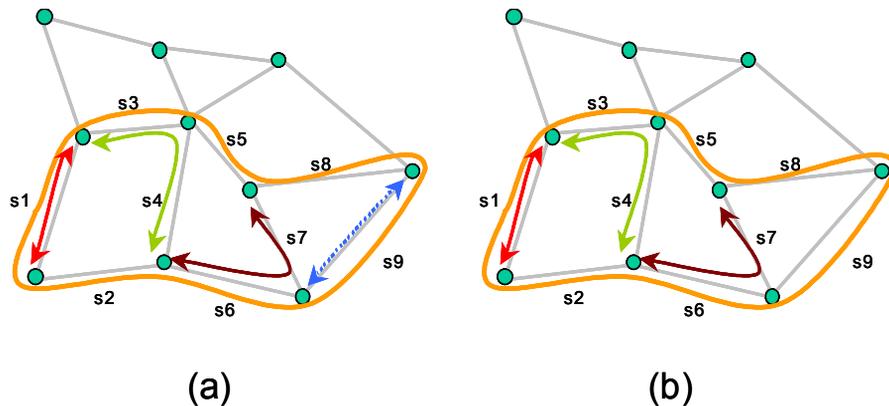
### 6.3.2 Overprotection and Actual Protection Assignment

The main output of the DRS method is the set of cycles that is needed to completely protect the demands against a single failure. The DRS method does not uniquely assign working paths to cycles, but only ensures that enough cycles are placed to guarantee 100% single-failure restorability. Because of the way that DRSs are generated, some working paths may have more protection relationships available to them than needed. The reason this happens is because sometimes the DRSs selected for protection by the solver include working paths that can be accounted for in other DRS/cycle configurations. For example consider two configurations A and B in which a particular demand relation is in a straddling and an on-cycle relationship, respectively. If this relation corresponds to a request of 2 units of demand, then it will be overprotected by A and B if both of the configurations are selected by the solver to be in the solution, which may happen if doing so does not negatively impact the total network cost.

In order to generate a visual solution of the problem, to perform dual-failure analysis, to perform transparent reach analysis, or in order to determine how many z-cases appear in the solution, every working path needs to be assigned to a specific cycle for its protection. In the case where there are exactly the required protection relationships available, this assignment is trivial. In the case where there are more protection relationships than required there may be several possibilities for actual protection

assignment. For the purposes of this study, unless stated otherwise, the bundle of unit working paths belonging to the same demand relation are thus assigned to cycles they straddle first, and only after all the straddlers are filled up are the on-cycle relationships considered. z-Cases, if there are any, are considered only after there are no more straddler or simple on-cycle relationships left.

The actual protection assignment described above may result in some cycles appearing to be longer than necessary for the DRS that they are protecting. This is also the case for the procedure that removes z-cases from configurations before they are passed to the solver. For example, Figure 6.1a shows a four route DRS (arrowed lines) and a cycle originally assigned to protect that DRS (the connected thick line) before assignment of protection. The dashed blue line represents a working path that has more than the required number of protection relationships provided by other DRS and cycle combinations (not shown). Figure 6.1b shows what may happen if the overprotected path does not end up being assigned to this particular cycle. The end result is that the cycle appears to unnecessarily go over spans s8 and s9 where it could have simply gone over s7 thus saving a unit of capacity.



**FIGURE 6.1.** A four route DRS (arrowed lines) and the cycle originally generated to protect the DRS (solid thick connected line). The dashed arrowed line represents the over protected working path.  
 (b) A possible outcome of actual protection assignment.

### 6.3.3 Common Test Network and Demand Data

The test network used in this study is the Germany Network [10], illustrated in Figure 6.2. In this figure two views are given: The first is of the network topology placed over top of a map of Germany with the major cities corresponding to the nodes of the network and the second is of the network topology where the spans and nodes are labeled as they will be referred to in the rest of the study. This network contains 17 nodes and 26 spans and has an average nodal degree of 3.06. This network contains a total of seven degree 2 nodes as well as two main hubs that are of degree 6 (Node 4, Hannover) and degree 5 (Node 11, Frankfurt).

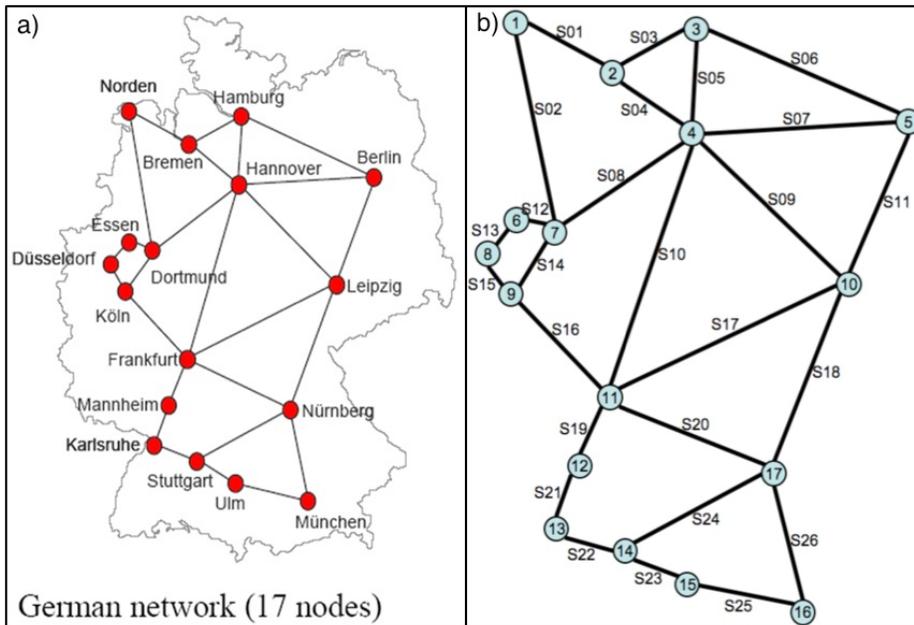


FIGURE 6.2. Germany test network. (a) City view. (b) Span & Node identification view.

The demand pattern used for this study is sparse. A total of 97 demand units are exchanged overall, with a total of 58 node-pairs exchanging demands out of a possible 136 node-pairs. A total of 37 node pairs exchange only a single unit of demand

and the highest volume exchanged between two nodes is 5 units. Specific information regarding the demand set can be found in the Appendix (Section 10.2.)

### 6.3.4 Experimental Setup

A total of four sets of experiments were run for the FIPP  $p$ -cycle architecture to generate minimum cost and minimum capacity solutions with and without allowing the  $z$ -case. Before the  $z$ -case was identified as a potential variant on the design concept, minimum capacity and minimum cost solutions allowing the  $z$ -case were generated using 30 DRSs per demand and 2 eligible cycles assigned to each DRS with the maximum DRSs size set to 11 routes. When it became apparent that the  $z$ -case had to be removed in order to comply with the nodal equipment assumptions, a second set of experiments was run. In these, minimum capacity and minimum cost solutions that did not allow the  $z$ -case were generated using 15 DRSs per demand, 4 eligible cycles assigned to each DRS and the maximum DRS size was set to 10 routes. The main idea behind the new parameters is that it was necessary to give the solver additional cycle options so that the impact of removing the  $z$ -case would not be as severe. Other parameters were reduced in order to compensate for the increase in problem size as the additional cycles were added.

Three test trials were run per test set using three different randomization seeds to generate different DRS problem instances. In minimum capacity designs, the eligible route set was generated by selecting a single least hop route for every demand. In the case where several working routes were the same number of hops, one was arbitrarily chosen. In the minimum cost designs, the eligible route set was generated by selecting a single least cost working route for every demand. Multiple demand units exchanged between two end-nodes were routed over the same spans, as opposed to being split over several equidistant routes. This criteria holds true for the working routing of all architectures under study, for comparison purposes. The FIPP DRS  $p$ -cycle model was implemented in AMPL 9.0 and solved using CPLEX 9.0. All experi-

ments were terminated with a mipgap of 0.01 (i.e., the solution may not be truly optimal, but the objective value is guaranteed to be within 1% of the optimal objective value). This software and mipgap value is applied to every experiment using an ILP in this chapter, unless stated otherwise.

For  $p$ -cycle, APS, DSP and PXT solutions generated as part of this study, the following methods and parameters were used. Optimal  $p$ -cycle solutions were generated using the ILP for generating minimum capacity optimal  $p$ -cycle solutions found in [35]. For this ILP, all possible cycles in the network were considered as eligible. DSP and APS solutions were generated using the DSP minimum capacity ILP method provided in [31]. APS solutions were generated using the same method by constraining the method to generate only two disjoint paths between every node-pair exchanging non-zero demand. PXTs solutions were generated by adapting the DRS method given in [40], which was originally written for in the context of FIPP  $p$ -cycles. The parameters were set such that every demand appeared in at least 30 DRSs, the maximum size of any DRS was set to 15 demand relations and 30 candidate trails were considered for every DRS.

### 6.3.5 Results

The results corresponding to the minimum capacity and minimum cost test sets where the  $z$ -case was not removed are in Table 6.1 and Table 6.2, respectively. Recall that capacity redundancy is defined as the spare capacity over the working capacity of the network. Cost redundancy is similarly defined as the cost of the spare capacity over the cost of working capacity of the network. The  $z$ -cases for both test sets are enumerated after the solution is returned by the solver. About 5% of all the demands in the network were found to be protected in a  $z$ -case relationship for each of the minimum capacity trials. This number was at most about 2% in the minimum cost cases and this difference is mainly attributed to the difference in working routing between the two test sets.

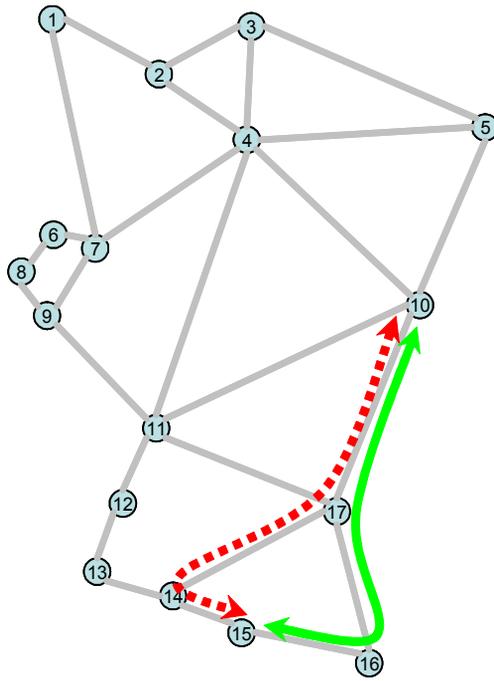
**TABLE 6.1. FIPP  $p$ -cycle single-failure restorable designs: Minimum capacity test set trials. z-Case not removed.**

<b>Trial</b>	<b>Working Capacity</b>	<b>Spare Capacity</b>	<b>Capacity Redundancy</b>	<b>Total # of Cycles</b>	<b>Total # of Unique Cycles</b>	<b># of z-Cases Protected (out of 97 demands)</b>
1	166	176	106%	15	10	5
2	166	176	106%	15	13	5
3	166	179	108%	15	11	5

**TABLE 6.2. FIPP  $p$ -cycle single-failure restorable designs: Minimum cost test set trials. z-Case not removed.**

<b>Trial</b>	<b>Working Cost</b>	<b>Spare Cost</b>	<b>Cost Redundancy</b>	<b>Total # of Cycles</b>	<b>Total # of Unique Cycles</b>	<b># of z-Cases Protected (out of 97 demands)</b>
1	23934	22300	93%	16	10	2
2	23934	22162	93%	16	8	1
3	23934	22056	92%	16	7	1

The results corresponding to the minimum capacity and minimum cost test sets where the z-case was removed are in Table 6.3 and Table 6.4 respectively. Most of the z-cases were removable using the method described above; by simply setting the  $x^{p,r}$  parameter to 0 whenever a z-case was detected. However, one demand in the network (between nodes 10 and 15) had to be deviated away from its shortest path as shown in Figure 6.3. The original shortest routing for this demand is shown using the dashed arrowed line which crosses nodes 10, 17, 14 and 15. The consequence of the fact that cycles are not allowed to cross the same node more than once is that any cycle protecting this demand must cross the span adjacent to nodes 14 and 15 as well as the span adjacent to nodes 10 and 17. This means that no cycle in the network can protect this demand without a z-case. In order to avoid this condition, the demand was re-routed as shown by the solid arrowed line to cross nodes 10, 17, 16 and 15. Working capacity wise, this did not make any difference because both routes are 3 hops long. In the minimum cost solutions, the cost of the working capacity increased from 23934 to 23974 when the condition that no z-cases may occur in the network was imposed.



**FIGURE 6.3. Modified working routing between nodes 10 and 15.**  
The dashed and solid lines represent the old and new route, respectively.

The spare capacity and cost of spare capacity in the non-z-case test sets, as shown in Table 6.3 and Table 6.4, increased compared to the test sets where the z-case was allowed. This was expected since some working paths could no longer be protected by the shortest cycles available. This forced the solver to pick longer cycles and more of them in order to fully protect all the demands in the network.

**TABLE 6.3. FIPP  $p$ -cycle single-failure restorable designs: Minimum capacity test set trials. z-Case removed.**

Trial	Working Capacity	Spare Capacity	Capacity Redundancy	Total # of Cycles	Total # of Unique Cycles
1	166	190	114%	18	12
2	166	192	116%	18	12
3	166	193	116%	16	13

TABLE 6.4. FIPP  $p$ -cycle single-failure restorable designs: Minimum cost test set trials. z-Case removed.

Trial	Working Cost	Spare Cost	Cost Redundancy	Total # of Cycles	Total # of Unique Cycles
1	23974	22772	95%	16	13
2	23974	23017	96%	17	10
3	23974	23377	98%	17	12

Up to this point both cost and capacity based solutions were presented for FIPP  $p$ -cycles. Since the ultimate goal of the study is to eventually focus on hardware issues such as line card placement, it makes sense to carry forward only the solutions that minimize the total number of hops. This is also supported by the fact that it is the cost of nodal equipment and not span-based cost that is of the most concern in this study. Furthermore, recall that it is assumed that the nodal equipment for this study is not capable of supporting the z-case. For these reasons, from this point on only the results from Table 6.3, corresponding to the minimum capacity non-z-case solutions, will be carried over to subsequent sections for further analysis.

The minimum cost and capacity results for all architectures under study are summarized in Table 6.5. The results presented are sorted in ascending order with  $p$ -cycles yielding the lowest cost and capacity solutions and APS having the highest capacity/cost. Note that the two columns correspond to two different test sets in which capacity and cost were minimized, respectively. The fact that APS had the highest cost of all the architectures was expected because in contrast to all other architectures it does not allow for any sharing of spare capacity. DSP does allow for sharing of spare capacity, but only provides marginal savings over APS according to the table. For this test network and demand pattern, DSP is unable to effectively exploit any spitting involving 3 paths or more. One reason for this is that more than half of the node pairs (37 out of 58) exchange only a single unit of demand forcing DSP degenerate into APS in these cases. This also happens because a large number (27 out of 58) demands have degree 2 end-nodes. Lastly, in the cases where 3 or more splitting options are

available for a set of end-nodes that are able to utilize it, most of the time the 3rd shortest path is much longer than the first and second one to the point where adding it is not cost effective.

By being able to more effectively provide spare capacity sharing, FIPP  $p$ -cycle and PXT solutions have lower costs than both APS and DSP. FIPP solutions are slightly better than PXTs as expected. The ability of FIPP to protect straddling routes gives this architecture an upper hand over PXTs by allowing a single protection structure to protect two units of working paths belonging to a single demand. In contrast, PXTs only allow for a single working path from any demand bundle to be protected by a single protection structure, meaning that more protection structures would have to be placed in the PXT case. Finally the optimal  $p$ -cycle ILP method yielded the lowest cost and capacity solutions.

**TABLE 6.5. Total capacities of the architectures under study.**

<b>Architecture</b>	<b>Total Capacity</b>	<b>Total Cost</b>
<i>p</i> -Cycles (optimal)	300	44198
<b>FIPP <i>p</i>-Cycles DRS method (no z-case, best trial)</b>	356	46746
<b>PXTs (DRS method)</b>	375	52849
<b>DSP</b>	445	63617
<b>APS (1:1 and 1+1)</b>	453	65199

## 6.4 Dual-Failure Restorability

The main goal of the next section is to consider how well single-failure restorable (R1) minimum capacity non-z-case designs from Section 6.3 perform in light of dual-failures. This was done by simulating dual span failures on every design and recording the number of failed working paths as well as the number of the paths that can be restored. Similarly to the previous section, it is assumed that the nodal equipment is simple and is only capable of detecting a failure and performing a single, pre-defined restoration action. Meaning that the nodal equipment does not have any

adaptive multi-failure capability: whenever a failure is detected on a working path, the same restoration response is attempted. After the network restores a single failure, the working paths affected by a second failure are not restored if any part of their backup paths are used or have failed.

In the following subsection, the results are presented using two metrics. The first metric for dual-failure restorability is the ratio of restored working path flows to affected (damaged) working path flows for all dual-failure scenarios, not counting the case where the two failures affect the same span. Failure order is not taken into consideration and the affected working path flows counter is incremented if either of the two failures affect the working path. If both failures affect the working path, the affected path flows counter is incremented by one. The second metric, intended to give a global view of dual-failure restorability, is the ratio of remaining functional flows after the restoration action takes place to the total number of flows in the network. The latter metric generally gives a much higher value than the prior because it considers all the flows in the network, not just the affected ones.

### **6.4.1 Method**

The dual-failure (R2) strategy used in this section works by first determining a single unique backup path for every working path based on the cycle assigned to protect that working path. If the working path is on-cycle, the backup path assigned to it is trivial; it is simply the part of the cycle that is disjoint from the working path. Every straddling working path has two potential backup paths available to it. If the straddler is carrying only one unit of demand, this working path is assigned the shorter of the two available backup paths. If the straddler is carrying two units of demand, both sides of the cycle are uniquely assigned as backup paths. Once these backup paths are established, it becomes possible to simulate all dual-failure scenarios and to determine the dual-failure restorability of the design. The set of all dual-failure scenarios does not include the case where the same span fails twice.

Operationally, in case of a single failure, the end-nodes of every affected working path perform a switching action that diverts the affected connection to its backup path. Upon a second failure, the end-nodes of the affected working paths only switch over to a backup path if it is intact and not currently being used to restore another connection. This strategy does not require any computing at the end-nodes nor does it require much signaling or synchronization between nodes. The only requirement is that the nodes are able to detect a loss of light, which acts as a trigger for the restoration mechanism.

There are two possible implementations of this strategy: with and without spare capacity release. The main focus will be on the strategy where spare capacity release is not allowed because it follows the assumption that the nodal equipment used is only capable of the simplest functions. This means that if the second failure affects a backup path that is being used to restore some working path from a first failure, the unaffected spare capacity used by the first restoration path is not released back to the network. If that spare capacity could be released, however, it may be possible to re-use that capacity to restore other affected working paths that would otherwise see their backup paths as being busy. The latter scenario is provided for comparison purposes only as it would not be implementable under the nodal equipment assumption of this study.

## **6.4.2 Results**

Table 6.6 and Table 6.7 show the results for R2 analysis without spare capacity release and with spare capacity release respectively. An approximate improvement of about 2% can be observed in R2 restorability (restored/affected) between the two cases, which is not high enough to justify the more complex equipment needed for spare capacity release.

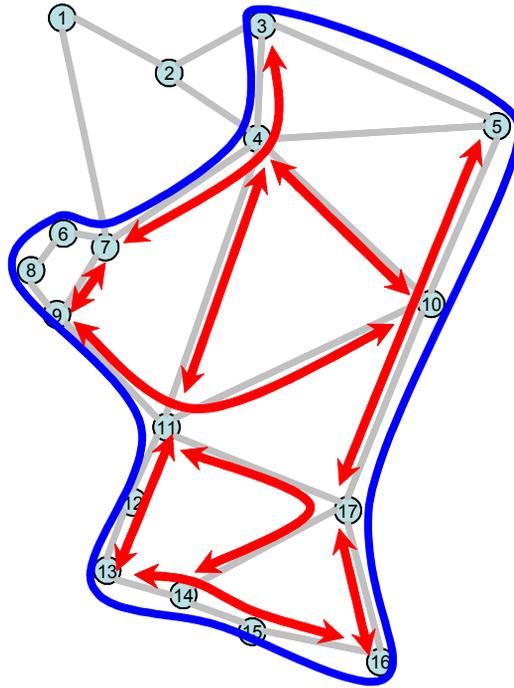
**TABLE 6.6. Minimum capacity non-z-case solution R2 analysis without spare capacity stub release.**

<b>Trial</b>	<b>Demands Affected</b>	<b>Demands Restored</b>	<b>R2 Restorability (Restored/Affected)</b>	<b>R2 Restorability (Functional/Total)</b>
1	8116	4310	53%	94.0%
2	8116	4149	51%	93.7%
3	8116	3985	49%	93.4%

**TABLE 6.7. Minimum capacity non-z-case solution R2 analysis with spare capacity stub release.**

<b>Trial</b>	<b>Demands Affected</b>	<b>Demands Restored</b>	<b>R2 Restorability (Restored/Affected)</b>	<b>R2 Restorability (Functional/Total)</b>
1	8116	4484	55%	94.2%
2	8116	4344	54%	94.0%
3	8116	4163	51%	93.7%

The results obtained indicate that approximately 50% of all affected demands will be restored after a second failure. This is a fairly low number considering the fact that redundancy of the single span failure restorable designs was shown to be over 100% in Section 6.3. The main reason for this can be attributed to the length of the backup paths used to protect the working paths in the network. The bulk of the demands are protected as part of large DRSs covered by correspondingly large cycles. Figure 6.4 illustrates one such large DRS and the cycle protecting it. The arrowed line segments indicate working paths, while the solid, connected line is the FIPP  $p$ -cycle. Consider what happens when a single failure affects the span between nodes 10 and 5. The affected on-cycle working path, which is originally only 2 spans long, is restored using a backup path crossing 13 spans. Note that at this point, not only is the original restored demand very vulnerable to a second failure, but the entire DRS that the demand was part of is also left completely unprotected. Basically, after the first failure, any failure affecting the spans crossed by the cycle shown (except for the span between nodes 17 and 10) or the spans crossed by any other working path in the DRS (except for the working path affected by the first failure) will result in un-restored service.



**FIGURE 6.4. Sample large configuration from the minimum capacity, non-z-case test set.**

Dual-failure restorability results for all the architectures under study are summarized in Table 6.8. Note that the results presented in Table 6.8 are in nearly exact inverse order the minimum capacity results presented in Table 6.5. Where DSP and APS solutions were the highest in terms of capacity use, they are now the highest in terms of dual-failure restorability. And where FIPP and  $p$ -cycle solutions had the lowest capacity use, they now also the lowest in terms of dual-failure restorability. The main effect seen here is that, because there is little or no spare capacity sharing in APS and DSP, these architectures are able to use dedicated protection paths that are very short or contain very few hops. This is beneficial in terms of dual-failure restorability because short protection paths have fewer failure points than longer ones. Additionally because the spare capacity is minimally shared the restoration of affected working paths often does not seize capacity needed by other working paths for restoration.

This is in contrast to architectures (such as  $p$ -cycles, FIPP  $p$ -cycles and PXTs) where sharing of capacity is used to reduce the overall capacity use, the consequence of which is that the protection paths end up being very long, relative to the working path. Furthermore, if several working paths are sharing the same capacity for restoration, the first failure and the subsequent restoration action renders the rest of the working paths unprotected from second failure. These two effects are particularly severe in the case of FIPP  $p$ -cycles, as was explained above. This is mirrored in the results where FIPP  $p$ -cycle minimum capacity solutions are shown to have the lowest dual-failure restorability.

**TABLE 6.8. Dual-failure restorability values the minimum capacity solutions for each architecture under study.**

<b>Architecture</b>	<b>R2 Restorability (Restored/Affected)</b>
<b>APS</b>	87%
<b>DSP</b>	86%
<b>PXTs (DRS method)</b>	67%
<b><math>p</math>-Cycles (optimal)</b>	65%
<b>FIPP <math>p</math>-Cycles (DRS method, no stub-release, best trial)</b>	53%

## 6.5 Minimum Wavelength Assignment

The next step in this study was to perform minimum wavelength assignment investigations. The intent was to examine how the different architectures may be implemented in a single fiber network where each wavelength may at most be used once in each fiber direction. The main assumption made is that nodes in the network do not have any wavelength conversion capability, meaning that every unit end-to-end path and every unit cycle needs to be assigned exactly one wavelength (per structure). This assumption is consistent with the main goal of this study, which is to investigate pre-connected architectures in the context of transparent optical networking.

This section is organized as follows: As an initial exercise, minimum wavelength assignment is performed on the minimum capacity single-failure restorable designs from Section 6.3. This is done to get a basic understanding how the different architectures compare to one another in this context. After the initial minimum wavelength exercise is complete, additional equipment constraints are added. It is assumed that the nodal equipment used to implement the designs is capable of supporting a total of 40 wavelengths, organized into two 20 wavelength bands. Under these constraints, the main goal is to make sure that no design exceeded the 40 wavelength limit. An additional goal, for the architectures where this is practical, is to see if it would be possible to generate designs that would need no more than 20 wavelengths. This second, more ambitious goal is motivated by the presumption that system costs could be reduced considerably if the designs could be implemented using only a single 20 wavelength band. For FIPP  $p$ -cycles, two approaches are used in order to try to achieve this goal. The first is to simply modify the minimum capacity ILP in order to generate solutions that use the fewest number of wavelengths possible. The second approach is to generate new minimum capacity designs using the FIPP DRS JCP ILP from Chapter 5 in order to reach the 20 wavelength goal.

### 6.5.1 Minimum Wavelength Assignment ILP Model

To optimally determine the minimum number of wavelengths needed to implement a minimum capacity FIPP  $p$ -cycle design, the following model was used:

**Sets:**

- $WL$             Set of wavelengths available.
- $S$                 Set containing the spans in the network.
- $D$                 Set of demand relations.

$D^s$	Set of demand relations that cross span $s$ . $D^s \subseteq D, s \in S$ .
$P$	Set of cycles.
$P^s$	Set of cycles crossing span $s$ . $P^s \subseteq P, s \in S$ .
$C$	Set of all DRSs.
$C^p$	Set of DRSs eligible for protection by cycle $p$ . $C^p \subseteq C, p \in P$ .
$C^r$	Set of DRSs that contain demand relation $r$ . $C^r \subseteq C, r \in D$ .

**Parameters:**

$d^r$	Number of demand units for demand relation $r$ . $r \in D$ .
$n^{p,c}$	Number of unit capacity copies of cycle $p$ needed to protect DRS $c$ . $p \in P, c \in C$ .

**Variables:**

$w^{\lambda,p,c}$	Equal to 1 if wavelength $\lambda$ is reserved for unit cycle $p$ protecting DRS $c$ and 0 otherwise. $\lambda \in WL, p \in P, c \in C$ .
$w^{\lambda,r,c}$	Equal to 1 if wavelength $\lambda$ is reserved for demand relation $r$ protected as part of DRS $c$ and is 0 otherwise. $\lambda \in WL, r \in D, c \in C$ .
$n^\lambda$	Equal to 1 if wavelength $\lambda$ is reserved for any path or cycle and is 0 otherwise. $\lambda \in WL$ .

## Min. WL Assignment ILP:

$$\text{Minimize: } \sum_{\lambda \in WL} n^{\lambda} \quad (6.1)$$

### Constraints:

$$\sum_{\lambda \in WL} w^{\lambda, p, c} = n^{p, c} \quad \forall p \in P, \forall c \in C^p \quad (6.2)$$

$$\sum_{\lambda \in WL} \sum_{(c \in C^r)} w^{\lambda, r, c} = d^r \quad \forall r \in D \quad (6.3)$$

$$\sum_{p \in P^s} \sum_{(c \in C^p)} w^{\lambda, p, c} + \sum_{r \in D^s} \sum_{(c \in C^r)} w^{\lambda, r, c} \leq 1 \quad \forall \lambda \in WL, \forall s \in S \quad (6.4)$$

$$n^{\lambda} \geq w^{\lambda, p, c} \quad \forall \lambda \in WL, \forall p \in P, \forall c \in C^p \quad (6.5)$$

$$n^{\lambda} \geq w^{\lambda, r, c} \quad \forall \lambda \in WL, \forall r \in D, \forall c \in C^r \quad (6.6)$$

The objective function (6.1) aims to minimize the number of unique wavelengths used. Constraints (6.2) and (6.3) ensure that every cycle and every unit working path is assigned a wavelength. Constraint (6.4) guarantees that individual wavelengths are used only once per span. No two unit structures, be it a cycle or a working path, can use the same wavelength on any span they share. The last two constraints, (6.5) and (6.6), make sure that any unique wavelength reserved for any cycle or working path is accounted for. The sum of the unique wavelengths flagged by the latter two constraints is minimized by the objective function.

### 6.5.1.1 Minimum Capacity Solutions Wavelength Assignment Results

The results of minimum wavelength assignment to minimum capacity non-z-case single-failure restorable designs from Section 6.3 are shown in Table 6.9. This number of wavelengths per trial is the minimum number of wavelengths that would

be required to support these designs. The value obtained also corresponds to the maximum single span total capacity for each design (also called the maximally loaded span). The minimum number of wavelengths cannot be lower than the capacity of the maximally loaded span, since all wavelengths crossing any single span must be different from one another.

**TABLE 6.9. FIPP  $p$ -cycle wavelength assignment results based on minimum capacity non-z-case solutions from Section 6.3.**

<b>Trial</b>	<b>Min. # of Wavelengths</b>	<b>Max. Single Span Total Capacity</b>
<b>1</b>	27	27
<b>2</b>	27	27
<b>3</b>	25	25

Recall that the minimum capacity solutions used in the previous initial exercise did not aim to reduce the total number of wavelengths used. Despite this fact, the wavelength assignment yielded a total of 25 wavelengths in the best case, according to Table 6.9, which is relatively close to being able to implement the designs using only 20 wavelengths. The next step in was to see how the DRS method could be modified (while still using the same working routing) so that the overall number of wavelengths used is decreased. To do this, the approach was to modify the DRS method so that the total capacity on any one span in the design was minimized. If the total capacity on any one span could be reduced to 20 or fewer units, it may be possible to implement this design using only 20 or fewer wavelengths as well. The rest of the section on minimum wavelength assignment will focus on this goal inline with trying to minimize the number of wavelengths used overall. Formally this was done by adding the following to the FIPP  $p$ -Cycle DRS SCP ILP (see Section 4.6 for more details):

**Parameter:**

$\nabla$  A small positive constant.  $\nabla = 0.001$ .

**Variable:**

$x$  Integer variable corresponding to the highest total capacity used on a span.  $x \geq 0$ .

**Minimize Max Total Single Span Capacity FIPP DRS ILP:**

$$\text{Minimize: } x + \nabla \cdot \sum_{i \in S} (s^i + w^i) \quad (6.7)$$

**Constraint:**

$$x \geq s^i + w^i \quad \forall i \in S \quad (6.8)$$

Constraint (6.8) keeps track of the most capacity used on any one span. This variable is minimized by the new objective function (6.7). Note that the new objective function includes a fractional term that ensures that the solver also minimizes the total capacity of the design to make sure that meaningful minimum capacity solutions are generated.

Table 6.10 shows the results obtained using the modified DRS ILP model. Note that for all three trials, the maximum total capacity on any one span is 23. This is the same as the minimum number of wavelengths required to implement these designs, as obtained using the minimum wavelength assignment ILP from Section 6.5.1. Recall that the lowest total capacity obtained from the minimum capacity non-z-case solutions from Section 6.3 was 356 units. Relative to this, the total capacity of these new designs increased only slightly (3% difference) in the best case. Note that to remain in line with the hardware assumptions, z-cases were removed from the new solutions using the same method as outlined in Section 6.3.

TABLE 6.10. New FIPP  $p$ -cycle wavelength assignment results.

<b>Trial</b>	<b>Total Capacity</b>	<b>Min. # of Wavelengths</b>	<b>Max. Single Span Total Capacity</b>
<b>1</b>	369	23	23
<b>2</b>	384	23	23
<b>3</b>	365	23	23

Figure 6.5a contains a histogram of the total capacity placement for every span for the solution from Trial 3 obtained using the modified DRS model above. This solution was selected as an example because it has the lowest total capacity out of all three trials. The histogram shows that the total single span capacity exceeds 20 units in only two cases while the rest of the spans are well below this threshold, averaging out to approximately 13.3 units. For illustration purposes, Figure 6.5b visually highlights the two spans where total capacity was 23. The fact that there are so few spans whose total capacity is above 20 and the amount head room available on other spans, suggests that it may be possible reduce the overall number of wavelengths used by deviating some of the working paths and cycles away from these spans. Deviating working and protection paths from these spans with the goal of generating solutions that can be implemented using a single 20 wavelength waveband is the focus of the following sections.

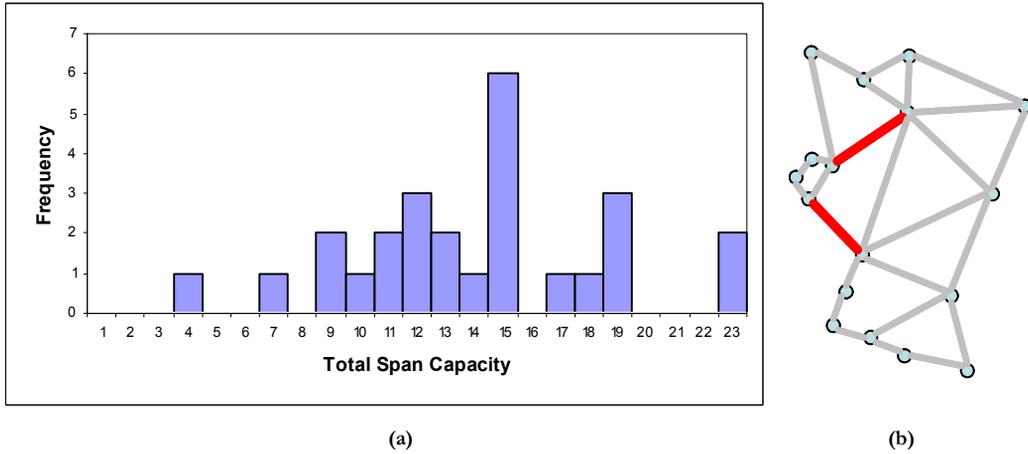


FIGURE 6.5. (a) Histogram of total capacity values and (b) illustration of spans where the total capacity was 23 units for the solution obtained in Trial 3 using the modified model.

## 6.5.2 Reaching a 20 Wavelength Solution

Recall, that the main reason for focusing on generating solutions that could be implemented using 20 or fewer wavelengths is because of the assumption that the network equipment is able to support 20 wavelength bands. Specifically, the equipment assumption here is that at each node, an multiplexer/demultiplexer (Mx/Dx) card is required for every waveband installed on each node per adjacent span. It is assumed that the main cost of the network comes from these cards and that fiber is costless. If the number of wavelengths used in a design could be kept at 20 units or below, only equipment necessary to support a single 20 wavelength band per span would have to be installed. The working paths and cycles could be can logically be arranged freely under this limit without additional cost. If the number of wavelengths needed to implement the design is only slightly higher than 20, additional equipment is required to be installed at every node crossed by the network structures needing to use the additional waveband. Since the minimum number of wavelengths reached so far is just over 20, there is motivation to try to reach exactly 20 wavelengths so that the total

number of wavebands is reduced. Doing so would prevent placing additional Mx/Dx and would therefore reduce the cost of the network.

An initial effort towards reaching the 20 wavelength threshold is to closely examine the minimum capacity solutions obtained using the modified DRS method to see if any cycles can be rerouted away from the spans exceeding 20 units of total capacity while keeping the working routing the same as was used in minimum capacity solutions above. It turns out that using this approach, 22 is the lowest number of wavelengths that could be reached. The reason behind it is illustrated in Figure 6.6 where it is shown that, using the minimum hops working routing, a total of 11 working paths cross the span adjacent to nodes 9 and 11. In this network, the only FIPP  $p$ -cycles capable of protecting these demands must also cross the same span which results in a minimum total capacity of 22 units for this span. This means that under the minimum hops routing used, it is not possible to generate a design that uses 20 or fewer wavelengths.

Demand n9-n10: 3 units  
 Demand n8-n11: 1 unit  
 Demand n9-n11: 5 units  
 Demand n8-n14: 1 unit  
 Demand n9-n14: 1 unit

---

Total: 11 units

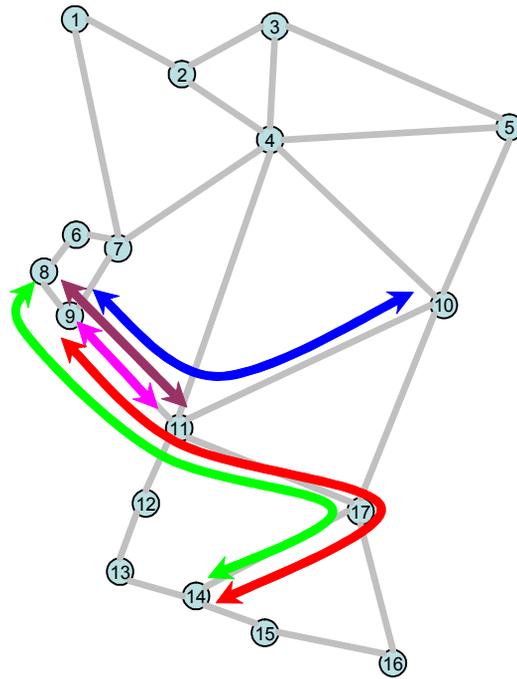


FIGURE 6.6. Illustration of the 11 units of demand crossing the span between node 9 and 11.

### 6.5.3 Joint DRS Model for FIPP $p$ -Cycles

In the previous section it was shown that it is not possible to generate a 20 wavelength FIPP  $p$ -cycle solution using the minimum capacity working routing generated for the initial exercise. In order to overcome this, a JCP model that places working capacity jointly with spare capacity was used to reach this goal. Specifically, the FIPP DRS JCP method outlined in Section 5.3 was used. This method allows for working paths to be deviated from the shortest and this is done by the solver only if doing so reduces the total network capacity. A new parameter, *routes per demand*, controls how many shortest routing options are added to the eligible route set, which is used to generate DRSs before they are passed to the solver.

#### 6.5.3.1 Experimental Setup

For this experiment the minimum number of DRSs that each demand was to appear in was set to 5 and 2 eligible cycles per DRS were considered. The maximum DRS size was set to 15 working routes and 3 minimum capacity eligible working routes were considered per demand. To keep in line with the assumption that the network equipment is only capable of simple switching actions, the z-case was preemptively removed using the same method as was used in Section 6.3. In order to make sure that the 20 wavelength goal was met, a constraint was added to the FIPP DRS JCP model. This constraint (6.9) makes sure that the total capacity on any one span does not exceed 20 units.

**Constraint:**

$$20 \geq s^i + w^i \quad \forall i \in S \quad (6.9)$$

**6.5.3.2 FIPP Joint Capacity Placement Results**

Using the FIPP DRS JCP model and the parameters specified, it was possible to obtain a 100% single-failure restorable solution that used a minimum of 20 wavelengths, as shown in Table 6.11. This was verified by running the minimum wavelength assignment model on the joint solution. Note that the working capacity has increased relative to the solutions obtained using the SCP methods where working capacity was 166 units. This increase was compensated for in the reduction of spare capacity, relative to the SCP solutions, as is expected for JCP methods. The total capacity of the joint solution remained relatively on par with the lowest (23) wavelength usage minimum capacity SCP solution obtained in the previous sections (where the total capacity was 365 units). Note that the number of cycles used decreased, relative to the minimum capacity solutions without wavelength restrictions. This is a reflection of the solver generating a more compact solution in an attempt to keep the maximally loaded span below 20 units, which implicitly allows for fewer wavelengths to be used.

TABLE 6.11. FIPP  $p$ -cycle 100% single-failure restorable JCP solution. Non-z-case, minimum capacity.

Working Capacity	Spare Capacity	Total Capacity	Total Number of Cycles	Number of Unique Cycles	Max. Single Span Total Capacity	Min. # of Wavelengths
182	187	369	12	7	20	20

The overall wavelength assignment results for all of the architectures under study are shown in Table 6.12. Note that in order to obtain these results, the SCP ILPs for PXTs and span-protecting  $p$ -cycles, as well as the DSP ILP with is necessarily a JCP method, were modified by adding a constraint which set the maximum total capacity for the maximally loaded span(s). Different values were tried until the lowest value was found for which the solver still returned a feasible solution. Only PXTs were able to reach the 20 wavelength threshold by using the modified SCP ILP. The joint method used for DSP was able to reach a minimum of 22 wavelengths, which is the fewest number of wavelengths possible for DSP for the demand pattern in the study. The lowest number of wavelengths that could be reached using the modified  $p$ -cycle SCP ILP was 22. The reason that span-protecting  $p$ -cycle SCP solutions cannot go lower than 22 wavelengths is the same as the reason given for FIPP  $p$ -cycle SCP solutions, as was explained in the previous sections. Specifically it is because the maximally loaded span uses 11 units of working capacity and these can only be protected using cycles that must also cross this span.

TABLE 6.12. Minimum wavelength assignment values the minimum capacity solutions for each architecture under study.

Architecture	Total Capacity	Minimum # of Wavelengths
FIPP $p$ -Cycles (DRS method, JCP)	369	20
PXTs (DRS method)	379	20
$p$ -Cycles (optimal)	317	22
DSP	445	22

## 6.6 Reach Limit Analysis

In an effort to make this set of studies more relevant to real world application, optical reach limits are considered in the following section. Recall that this study focuses on fully pre-connected architectures in the context of optical networking. The assumption made is that the equipment used is not capable of electrical signal regeneration and that signals enter the optical domain as they enter the network and remain there until they are terminated at their destination node. While in the optical domain, the signals are subject to effects such as attenuation and dispersion from the fiber as well as insertion losses at the intermediate nodes. The network is only able to transmit signals a certain distance before these effects degrade the signal below acceptable levels.

Where transmission paths pass through transparent nodes, an insertion loss of 80km is added to the transmission distance of that path. The total transmission distance for an end-to-end path, also known as transparent reach, is determined by equation (6.10) where  $R$  is the total transmission distance,  $h$  is the number of hops of the path being considered,  $i$  is the insertion loss and  $L$  is the sum of the geographical lengths of the spans crossed by the path. For this study, it is assumed that the maximum transmission distance for any end-to-end path is 2000km.

$$R = (h - 1) \cdot i + L \quad (6.10)$$

Essentially the main goal of this section is to generate a minimum capacity solution that is also transparent reach feasible. In other words, a solution where no path, working or protection, exceeds the maximum transmission distance. As an initial step, previously generated solutions are examined to see if they are reach feasible. Specifically the two solutions analyzed are Trial 1 of the minimum capacity non-z-case solution from Section 6.3 as well as the joint 20 wavelength minimum capacity solu-

tion from Section 6.5. After the initial step is complete, a new reach feasible section is generated and contrasted against the solutions from previous sections.

### 6.6.1 Analysis of a Minimum Capacity Solution

Figure 6.7 shows the transparent reach of every unit working and backup path from Trial 1 of the minimum capacity non-z-case solution from Section 6.3. Specifically the figure shows the distribution of the reaches of the working and protection paths. None of the working paths (illustrated by the light bars) exceed the transparent reach limit. The transparent reach of the longest working path does not exceed half of the threshold of 2000 km. In contrast, 45.4% of protection paths (illustrated by the dark bars) exceed the transparent reach limit with the longest protection path being nearly 60% longer than the maximum transmission distance. Clearly, this minimum capacity solution is not reach feasible. The main reason for this is that most demands in this solution are protected by very large cycles. The protection paths these cycles provide end up being many times longer than the working paths themselves.

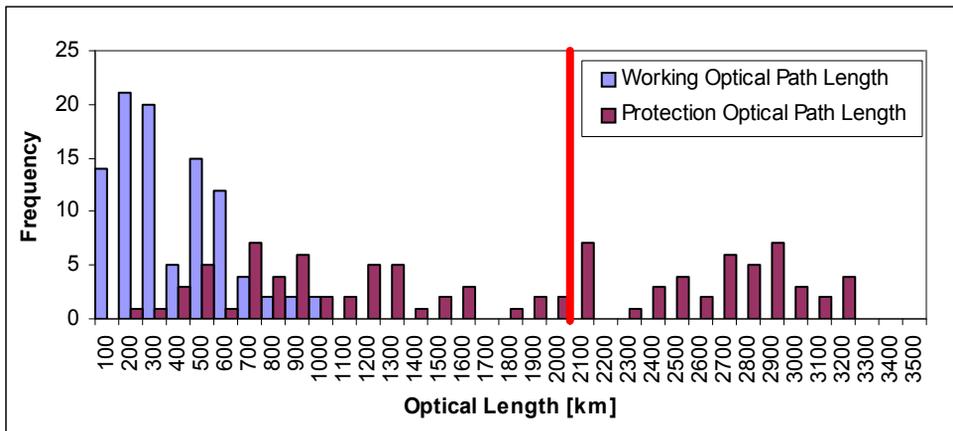


FIGURE 6.7. Transparent reach analysis: Minimum capacity solution (Trial 1, non-z-case.)

## 6.6.2 Analysis of a Minimum Capacity Under 20 Wavelengths Solution

Figure 6.8 shows the transparent reach distribution of the unit working and protection paths of the joint 20 wavelength minimum capacity solution from Section 6.5. Despite being able to deviate from the shortest working routes, the working paths (light bars) remain very short and fall far short of the 2000 km transparent reach limit. This effect is expected of joint solutions where the working paths are deviated very slightly from the shortest working paths (when optimal) in an effort to reduce the total capacity used or, in this case, to reduce the total number of wavelengths needed to implement the solution. The number of protection paths that exceed the maximum transmission distance is 57.7%, which is approximately a 12% increase relative to the SCP solution. This effect can be explained by looking closer at the numbers and sizes of cycles used in the JCP and the SCP solutions. The SCP solutions contained some large and some medium cycles. In contrast, the JCP solution contained fewer cycles over all (so that the number of wavelengths used could be reduced) but all of these cycles were very large causing the number of protection paths whose transmission reaches exceeded the maximum transmission distance to increase, relative to the SCP case.

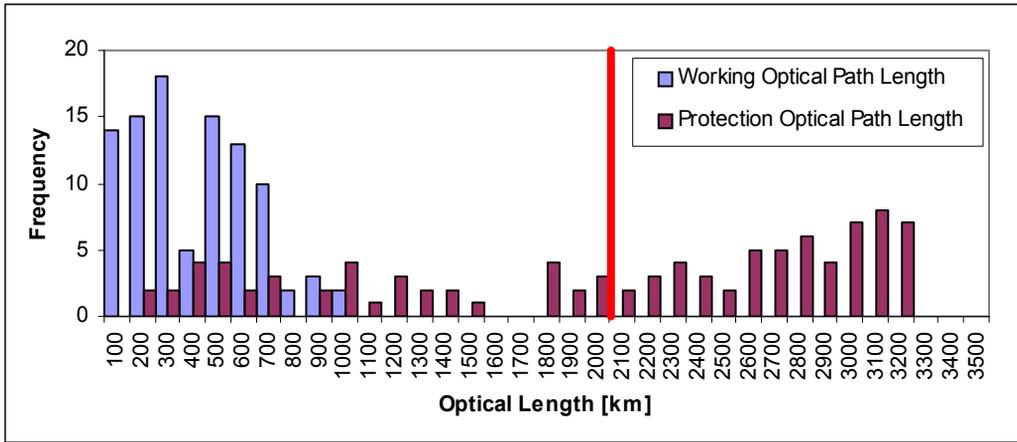


FIGURE 6.8. Transparent reach analysis: Minimum capacity, JCP, 20 wavelength solution.

### 6.6.3 Reach Limit Feasible Solution

At this point, it is clear that the minimum capacity solutions generated thus far are not adequate when it comes to transparent reach. In order to generate a solution that is transparent reach feasible, an approach similar the z-cases removal method was used. This method involves adjusting the data file before it is passed to the solver in a way that prevents the solver from using protection paths that exceed the maximum transmission distance. Recall that the data file contains a parameter  $x^{p,r}$  that is set to 2 or 1 depending on whether cycle  $p$  and demand relation  $r$  are in a straddling or an on-cycle relationship, respectively. This parameter is set to 0 if cycle  $p$  cannot be used to protect demand relation  $r$ . This parameter can also be interpreted as the number of protection paths provided by the cycle to the demand relation. To generate a solution that is reach feasible, this parameter was modified to represent the number of protection paths provided by the cycle that do not exceed the maximum transmission distance. Once this was done, the FIPP DRS ILP model from Section 4.6 was used to generate a reach feasible solution. The ILP parameters were such that every demand

relation appeared in at least 15 DRSs, 10 eligible cycles were considered for each DRS and the size of the DRSs was limited to 6 working routes. The latter parameter was set deliberately low to ensure that the DRSs remain small and are protected by small cycles in an effort to reduce the transmission distances of protection paths.

The results of the experiment are shown in Table 6.13. As expected of all SCP solutions, the working capacity remained at 166 units. The spare capacity used increased by 41 units, corresponding to a 25% capacity redundancy increase relative to the best minimum capacity non-z-case solution from Section 6.3. The overall number of cycles used also increased. This was expected since the solver was no longer able to protect large DRSs and was forced instead to protect numerous small DRSs using smaller cycles. Figure 6.9 shows the transparent reach distribution of the unit working and protection paths for the reach limit feasible solution. It shows that none of the working or protection paths exceed the reach limit as was desired.

**TABLE 6.13. FIPP  $p$ -cycle 100% single-failure restorable design:  
Minimum capacity, non-z-case, transparent reach feasible solution.**

<b>Working Capacity</b>	<b>Spare Capacity</b>	<b>Capacity Redundancy</b>	<b>Total Number of Cycles</b>	<b>Number of Unique Cycles</b>
166	231	139%	28	23

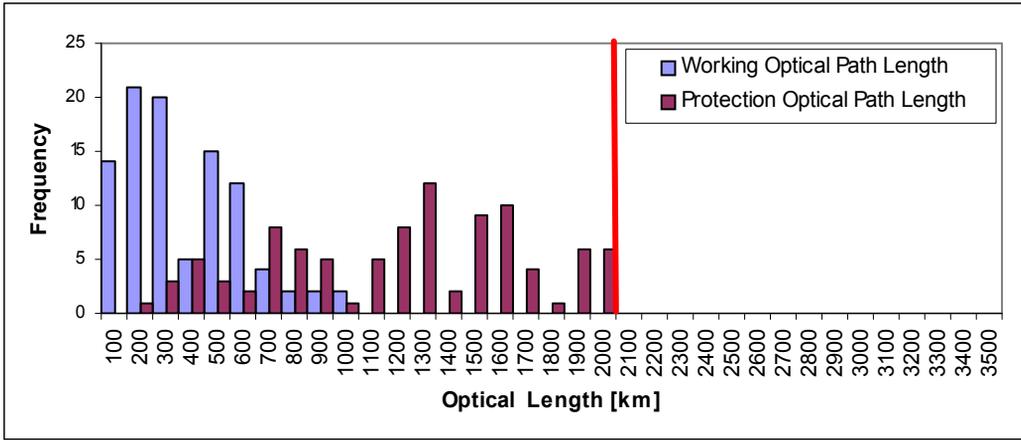


FIGURE 6.9. Transparent reach analysis: Minimum capacity, reach feasible solution.

### 6.6.4 Overall Reach Limit Results

The total capacities of the reach feasible solutions for all architectures are shown in Table 6.14. For  $p$ -cycles, a new ILP was developed for this study that was capable of optimally generating solutions while making sure that the recovery paths used for every span failure did not exceed the maximum transmission distance. It turns out that this new model was able to generate a reach feasible solution without requiring any additional capacity over the minimum capacity solutions generated in Section 6.3. The details regarding this model can be found in [75]. The method outlined above for FIPP  $p$ -cycles was essentially the same one used when it came to generating reach feasible PXT solutions. In the case of PXTs, the 20 wavelength constraint was also added to see if both it and the reach limit constraints can be satisfied simultaneously, with great success. In the case of DSP, the original minimum capacity solutions turned out to be reach feasible and no additional work had to be done. The reason for this is because the spare capacity in DSP solutions is only sparsely shared, allowing for the shortest or near shortest working and protection paths to be used.

TABLE 6.14. Total capacities of reach feasible solutions for each architecture under study.

Architecture	Total Capacity
<i>p</i> -Cycles (optimal)	317
PXTs (DRS method)	386
FIPP <i>p</i> -Cycles (DRS method)	397
DSP (optimal)	445

## 6.7 Node Failure Restorability

Up to this point, the main focus of the previous sections was to generate minimum capacity designs that were 100% single span failure restorable. The focus on span failures was motivated by the fact that span failures are the most common type of failure in transport networking. The main reason for this is that spans represent point-to-point fiber optic links that are easily damaged by digging, animals, natural disasters and accidents. Another type of failure that can occur in a network is a nodal failure. This type of failure is much more rare, considering that each node represents entire switching offices where the equipment is housed in secure rooms with redundancy built into the transmission and power systems and that have maintenance staff making sure that things run smoothly. However, given the possibility of fire, a severe power outage, failure of core components, accidents and human error it is still possible for a network to experience an entire node failure. It is therefore of interest to investigate the effect such a failure would have on the operation of the network.

The following section is dedicated to nodal failures and how they affect FIPP *p*-cycle designs. First, minimum capacity single-failure designs from Section 6.3 are subjected to node failures to determine how resilient these networks can be without any preemptive consideration. The intent is to determine the extent of inherent nodal R1 failure resilience occurring in FIPP *p*-cycle minimum capacity designs. The next step is to examine what can be done given these designs after the failure has occurred so that the available resources are used in an efficient manner. Initially, a first come

first serve approach is taken and is followed by the development of a method that optimally allocates the available resources after a node failure, maximizing restorability. Finally a modified DRS method is proposed that generates FIPP  $p$ -cycle designs that are 100% resilient against single node failures.

### 6.7.1 At-a-Glance Analysis

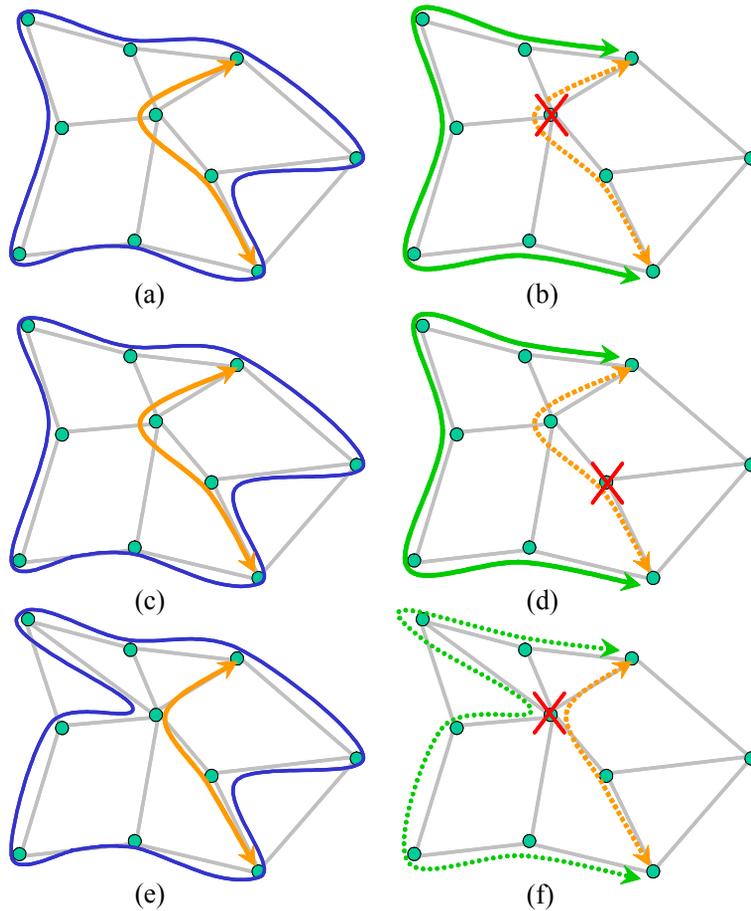
Before any nodal R1 analysis or design for 100% nodal R1 can take place, it is useful to take a closer look at how FIPP  $p$ -cycle designs respond to single node failures. It should be noted that in the case of any path-protecting network architecture, it is assumed that no restoration is possible for demand relations whose end-nodes have failed. In this context, 100% nodal R1 refers to the ability of the design to protect working paths from failures occurring at the intermediate nodes they cross. The following subsections illustrate the ways in which node failures may affect FIPP  $p$ -cycles that were originally designed with only single span failures in mind.

### 6.7.2 Node Failures and On-Cycle Relationships

Figure 6.10 illustrates three different ways in which a node failure can affect a single route that is in an on-cycle relationship with the protection cycle. Figure 6.10 (a, c and e) illustrate the cycle and the working route protected by it before the failure occurs. In these figures, the working route is represented by the arrowed line and the cycle is represented by the solid connected line. Figure 6.10 (b, d and f) show the network state after a node failure, the location of which is represented by the large X. The dashed arrowed line represents the failed working route while the solid arrowed line represents the enabled protection path.

Figure 6.10 (a and b) illustrate that when a node failure severs an individual partially on-cycle route but does not affect its protection cycle, the reaction that takes place is the same as would occur in the case of a single span failure. The default pro-

tection path (unaffected by the node failure) is enabled and the demand relation is restored. Recall that, because the equipment being used is very simple, each working path is assigned a default protection path that is enabled in case of a failure. A similar reaction takes place if the cycle as well as the working route are affected, shown in Figure 6.10 (c and d), as long as the failure does not affect the default protection path of the failed working route. However, if the node failure does affect the working route as well as the protection path at the same time, as illustrated in Figure 6.10 (e and f), the default protection path can no longer be used to recover the failure. Note that in this case there is actually an unaffected protection path that connects the end-nodes of the failed demand relation. However, this path is not the default protection path and may not be used in this situation because the equipment assumed in this study would not be able to support it. This case is similar to the *z*-case relationship encountered in span restorable designs where it was not possible to get 100% restorability if these configurations were left in the network designs. Similarly, these ‘nodal *z*-cases’ will have to be avoided in addition to the ‘span *z*-cases’ in order for the solutions to be 100% node failure restorable.

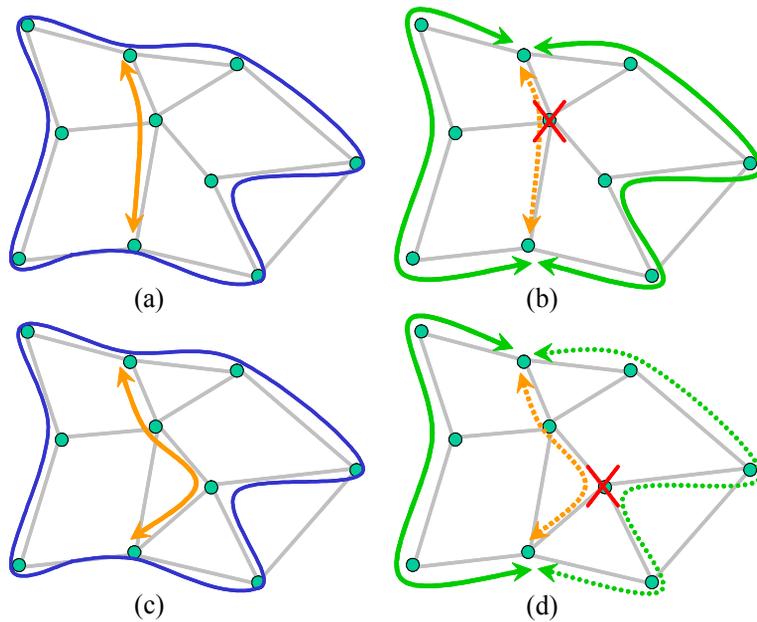


**FIGURE 6.10. Nodal failures affecting on-span working routes.**  
 (a,c,e) represent the network state before failure and (b,d,f) represent  
 the state of the network after failure.

### 6.7.3 Node Failures and Straddling Relationships

Using the same notation as in the on-cycle section above, Figure 6.11 illustrates the possible ways that a node failure may affect a working route in a straddling relationship. Recall that this type of a relationship allows for the protection of 2 working paths over this type of route against a single span failure by being able to break into both sides of the cycle at the end-nodes. Figure 6.11 (a and b), similarly to the on-cycle case, illustrate that when a node failure affects only the working route but not

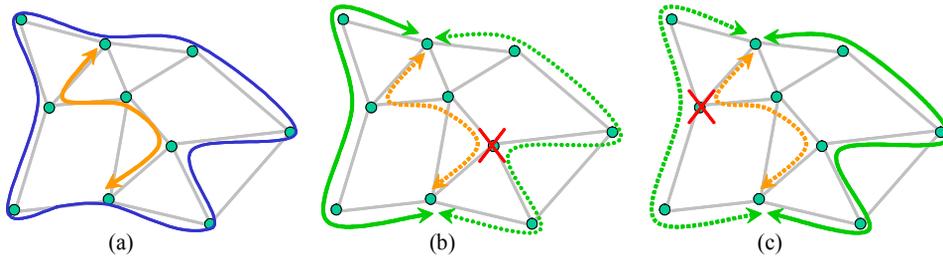
the cycle, the reaction is the same as if a single span failure had occurred. Both the working paths on this failed route are recovered. However, if the straddling route is not node disjoint from the cycle and the node shared by the cycle and the route fails, as illustrated by Figure 6.11 (c and d), then only one unit of working capacity may be restored. In other words, a straddler that is node disjoint from the protection cycle may provide 2 units of protection to the working paths routed on it. If the straddler is not node disjoint from the cycle, then the cycle can provide at most a single unit of protection.



**FIGURE 6.11. Node failures affecting straddling working routes.**  
 (a,c) represent the network state before failure and (b,d) represent the network state after failure.

Another type of relationship possible when it comes to protecting a single straddling route is the one shown in Figure 6.12a. In this case the straddling route is not only not node disjoint from its protection cycle, but it is not node disjoint from any of the possible protection paths provided by the cycle. This is illustrated by the two failure states in Figure 6.12 (c and d) where the protection paths provided by the

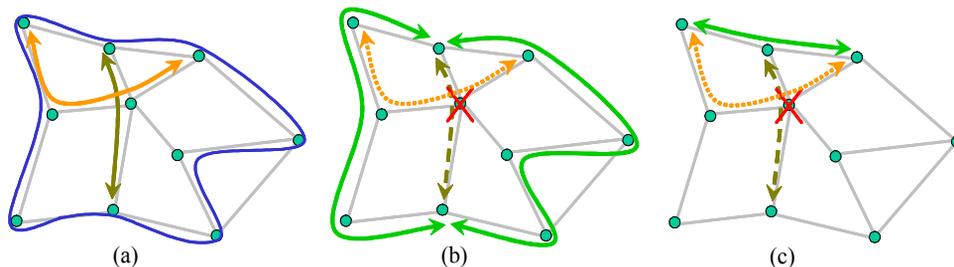
cycle fail simultaneously with the working route. This is the straddler version of the ‘nodal z-case’ and it will not yield 100% single node failure restorability using the equipment assumed in this study.



**FIGURE 6.12. Nodal z-case: a straddling working route that is not node disjoint from the protection paths provided by the cycle. (a) is the network state before failure and (b,c) are examples of possible network states after failure.**

#### 6.7.4 Mutual Capacity Problem

The scenarios in the two previous subsections only considered single working routes and how they are affected by single node failures. Additional consideration needs to be taken with regard to the fact that most cycles are shared by more than one working route. Figure 6.13 illustrates a situation that would have otherwise been allowed in the minimum capacity single span failure restorable designs. The working routes in Figure 6.13a can simultaneously be affected by a node failure as shown in Figure 6.13 (b and c). When this happens, the two failed working routes compete for the spare capacity provided by the cycle and the order in which they acquire this capacity has a strong effect on the nodal restorability of the design. If the straddling route gets priority, 2 of the 3 failed working paths can be restored as shown in Figure 6.13b. However if the on-cycle route is restored first, as in Figure 6.13c, then only 1 of the 3 failed working routes are restored, resulting in poorer restorability. This type of spare capacity contention is usually referred to as a mutual capacity problem.



**FIGURE 6.13. Mutual capacity scheduling conflict occurring as a result of node non-disjoint working routes sharing the same cycle for protection. (a) is the network state before failure and (b,c) are examples of possible network states after failure.**

There are two main insights to take away from the cases presented in the previous three subsections relating to whether a design can be 100% node failure restorable or not. First, it is important to make sure that the working routes that share a cycle for protection are node disjoint from one another. This will ensure that no two working routes can fail simultaneously and thus can never compete for the same spare capacity after a node failure. Second, any working path seeking protection against single node failures (whether part of a straddler or an on-cycle route) must be node disjoint from its default protection path.

### 6.7.5 Single Node Failures in Single Span Failure Restorable Minimum Capacity Designs

The main goal of this section is to determine how well single span failure restorable (R1) minimum capacity non-z-case designs from Section 6.3 perform in light of node failures. These solutions were generated in three trials using the FIPP DRS method. For these designs, the biggest problem in the context of node failures is the mutual capacity problem as described above. This occurs when three or more working paths that share the same protection cycle are affected by a node failure. Fewer than three working paths failing at the same time do not encounter the mutual capacity issue. If only a single path is affected by a node failure, restoration action is

trivial: the end-nodes of the working path break into the cycle and enable the protection path, if it is still intact. If two working paths are affected, and both of their protection paths are intact but cross some of the same spans, then one is selected for restoration arbitrarily.

The following two sections introduce two methods that can be used to deal with the mutual capacity problem. Both of these methods simulate every possible node failure for every DRS and its respective FIPP  $p$ -cycle. Recall that a DRS, in this case, is a set of span disjoint working routes from the minimum capacity single span failure restorable solutions that share protection under a cycle. Additionally, in line with the equipment assumptions made at the beginning of this study, every working path is assigned a single protection path that is automatically switched to in case the working path fails.

### **6.7.6 First Come First Serve Method**

As a first approach to the mutual capacity problem, the first come first serve (FCFS) method can be used to arbitrate the selection of the enabled protection paths when more than one working path is affected by a node failure. Whenever there is a conflict between protection paths, using the FCFS method, they are simply picked and enabled one at a time. If the resources that the path needs are already used up then the path is ignored and is presumed not to be recoverable. The order that the conflicting paths are selected is arbitrary. Implicitly, this method can be used to generate the average level of restoration for a particular design.

### **6.7.7 Optimal Conflict Resolution Method**

The second approach optimally selects which protection paths are enabled, with the goal of maximizing the total number of restored working paths. This is done using the Optimal Nodal R1 Conflict Resolution model below. Note that this model

is called for each DRS/cycle configuration. Multiple instances of a configuration are considered simultaneously by this model. The results from this model for every configuration are summed together to obtain the total number of restored working paths. Before the model can be run, the protection paths of the working paths affected by the failure are placed into set  $A$  as unit structures. The model then decides the optimal set of enabled protection paths based on the resources available (represented by  $\beta$ ).

**Sets:**

- $S$  Set containing the spans of the network.
- $A$  Set of unit protection paths seeking restoration. This set is generated for every DRS considered based on the working paths affected by failure.

**Parameters:**

- $x^{\alpha, i}$  Equal to 1 if protection path  $\alpha$  crosses span  $i$  and is 0 otherwise.  
 $\alpha \in A, i \in S$ .
- $\beta^i$  This parameter encodes the amount of spare capacity available on span  $i$ . This value is obtained from the number of cycles protecting the DRS being analyzed.  $i \in S$ .

**Variables:**

- $m^\alpha$  Equal to 1 if protection path  $\alpha$  was selected to be used for restoration and is 0 otherwise.  $\alpha \in A$ .

$m^{\alpha, i}$  Equal to 1 if protection path  $\alpha$  crosses span  $i$  and is 0 otherwise.  
 $\alpha \in A, i \in S$ .

### FIPP\_OptNodalR1\_ConflictResolution\_ILP:

$$\text{Maximize: } \sum_{\alpha \in A} m^{\alpha} \quad (6.11)$$

#### Constraints:

$$m^{\alpha} = m^{\alpha, i} \cdot x^{\alpha, i} \quad \forall \alpha \in A, \forall i \in S \quad (6.12)$$

$$\sum_{\alpha \in A} m^{\alpha, i} \leq \beta^i \quad \forall i \in S \quad (6.13)$$

The objective function (6.11) maximizes the number of protection paths that get selected to restore their respective working paths. Constraint (6.12) ensures that if a protection path  $\alpha$  is selected to be enabled for restoration, then the resources (spare capacity on the spans it crosses) used by this path are accounted for. Constraint (6.13) makes sure that the resources used by the protection paths selected do not exceed the spare capacity provided by the cycle.

### 6.7.8 Results

The nodal failure analysis results obtained for the minimum capacity 100% single span failure restorable designs for FIPP  $p$ -cycles are provided in Table 6.15. Note that the 194 unit working path failures that correspond to unrestorable end-node failures are not included in Table 6.15. This number is twice the total number of demands, corresponding to the failures of the demands' respective end-nodes. The table only provides the results for affected transiting demands.

The first column of Table 6.15 provides the total number of demand units affected by node failures not counting end-node failures. This number (69) is the same for all architectures and trials that used the same least capacity routing. The ‘restorable before capacity contention’ column contains the total number of protection paths that are eligible for recovery from the node failures. The percentile values are relative to the number of transiting affected paths. The reason that this number is lower than the number representing the transiting affected paths is because some working paths were in nodal z-case configurations with their protection cycles meaning that their protection paths failed along with the working paths during the simulated node failures, rendering those demands unrestorable. The difference between the total number of transiting affected paths and the number of potentially restored working paths is the measure of how many ‘nodal z-cases’ came up in the single span failure solutions. The potentially restored value represents the number of surviving protection paths that have a chance of being selected to restore their respective working paths. This is the maximum number of demand units that can be restored before the mutual capacity problem is taken into consideration.

The next column contains the number of demands restored using the FCFS method. Using this method it was possible to get nodal R1 in the range of 74% to about 81%. The last column provides the nodal R1 values obtained using the optimal conflict resolution method. It ranged from 74% to 84%. Note that this method was only able to save an additional two working paths in the first two trials and it turns out that using the FCFS method yielded the optimal solution in the Trial 3. While the values for the FCFS and the Optimal Conflict Resolution method were very close for this network, there may be greater disparity between the two if this experiment was run on a higher degree network because there would be more instances of a large number of working paths failing simultaneously.

**TABLE 6.15. Minimum capacity single span failure solutions:  
Comparison of results obtained using the FCFS and the Optimal  
Conflict Resolution method.**

<b>Trial</b>	<b>Transiting Affected Paths</b>	<b>Restorable Before Capacity Contention</b>	<b>Actually Restored Using FCFS</b>	<b>Optimally Restored</b>
1	69	63 (91%)	56 (81%)	58 (84%)
2	69	58 (84%)	53 (77%)	55 (80%)
3	69	55 (80%)	51 (74%)	51 (74%)

## 6.8 100% Node Failure Restorable Designs

The main goal of this section is to present an algorithm for generating 100% R1 node failure restorable designs and to investigate the capacity cost required to achieve this protection. In order to formally define the requirements necessary for 100% node failure restorability, the following sets and parameters must be introduced:

### Sets:

$N$  Set containing the nodes of the network.

$C$  Set of all protected path sets.

$Q$  Set of all working paths.

$Q^c$  Set of all working paths in the protected path set  $c$ .  $Q^c \subseteq Q, c \in C$ .

$B^{q,c}$  Set of the protection paths available for working path  $q$  as part of the protected path set  $c$ .  $q \in Q^c, c \in C$ .

### Parameters:

$z^{n, q, c}$  Equals to 1 if node  $n$  is crossed by working path  $q$  as part of the protected path set  $c$  and is 0 otherwise.  $n \in N, q \in Q^c, c \in C$ .

$z^{n, b, c}$  Equals to 1 if node  $n$  is crossed by the protection path  $b$  as part of protected path set  $c$ .  $n \in N, b \in B^{q, c}, q \in Q^c, c \in C$ .

Based on the previous subsections and definitions above, for a FIPP design to be 100% R1 node failure restorable the following must be true:

**Constraints:**

$$\sum_{q \in Q^c} z^{n, q, c} \leq 1 \quad \forall n \in N, \forall c \in C \quad (6.14)$$

$$z^{n, q, c} + z^{n, b, c} \leq 1 \quad \forall b \in B^{q, c}, \forall q \in Q^c, \forall c \in C, \forall n \in N \quad (6.15)$$

Where by constraint (6.14) all working paths in the protected set must be node disjoint from one another and, by constraint (6.15) the working paths must also be node disjoint from their backup paths. The following section outlines the procedure used to meet these constraints.

### 6.8.1 Experimental Setup and Assumptions

As with the minimum capacity solutions, the FIPP DRS method can be used to generate the 100% node restorable solutions. However in this case, the DRS method has to be modified in order to fulfill the node failure restorability criteria.

First, minimum capacity working routes are generated for each demand. It is then checked whether the routes are in a nodal z-case relationship with every possible cycle in the network, and if this is the case, the routing is altered to avoid this situa-

tion. This type of a z-case relationship is referred to as a topological nodal z-case. This occurs because the network topology is such that the placement of a route in question and the constraint that only simple cycles (nodes are only allowed to be crossed once) are to be used as protection structures cause that route to be in a nodal z-case relationship with every cycle eligible to protect it. Recall that if a route is in a nodal z-case relationship with a cycle, it means that neither of the two possible protection paths provided by the cycle are node disjoint from the working route. This means that, given the hardware assumptions in this study where a single pre-determined backup path is selected for restoration, both the protection path and the working route can fail at the same time. Since this would result in a restorability of less than 100%, the routes that are in a nodal z-case relationship are avoided.

Once working routing is established, DRSs are built using a DRS generation algorithm that is modified to ensure that the all the routes in the set were node disjoint from one another. Finally, the cycle-route specific z-cases are removed by modifying the  $x^{p,r}$  parameter to correctly reflect the number of node disjoint protection paths provided to each route by each eligible cycle. Specifically,  $x^{p,r} = 2$  if there are two node disjoint protection paths available for demand relation  $r$  given cycle  $p$ ,  $x^{p,r} = 1$  if there is just one node disjoint protection path provided and  $x^{p,r} = 0$  otherwise. As input to the problem, it is ensured that each demand appeared in at least 15 DRSs. 4 eligible minimum capacity cycles are assigned to each DRS for protection and the maximum DRS size is set to 10 routes.

## 6.8.2 Results

Table 6.16 shows the costs of solutions obtained under the constraint of 100% node failure restorability. Working capacity, as shown in the first column, increased by two units relative to the minimum capacity non-z-case solutions from

Section 6.3 (which was 166 units). This was the result of the working routing being deviated from the minimum capacity routing in order to avoid topological nodal  $z$ -cases. The spare capacity, standard redundancy and the number of cycles also increased due to the fact that the problem is much more constrained compared to the minimum capacity single span failure problem. The standard redundancy increased by  $\sim 20\%$  relative to the span failure restorable solution. In this case standard redundancy was used in order to compensate for the increase in the working capacity. Recall that standard capacity is the ratio of total capacity less the minimum working capacity possible, all divided by the minimum working capacity possible. The number of cycles used also increased proportionally to the spare capacity used.

**TABLE 6.16. FIPP  $p$ -Cycle Single Node Failure Restorable Designs: Minimum Capacity Test Set. Nodal  $z$ -cases removed.**

<b>Trial</b>	<b>Working Capacity</b>	<b>Spare Capacity</b>	<b>Standard Capacity Redundancy</b>	<b>Total # of Cycles</b>	<b>Number of Unique Cycles</b>
<b>1</b>	168	222	135%	22	18
<b>2</b>	168	215	131%	20	15
<b>3</b>	168	217	132%	22	15

In order to verify that the solutions were indeed 100% node failure restorable, the Optimal Conflict Resolution method was used on each trial. The results of this are shown in Table 6.17. The first column contains the number of transiting affected paths over all node failures. The second column contains the number of paths restorable before contention is considered. Recall that contention is used here to describe the effect of several working paths seeking to use the same spare capacity on the same cycle. It was previously shown that when contention exists, the number of paths restored depends on the order in which they are allowed seize the cycle's resources. Since nodal  $z$ -cases were removed, the first two columns contain the same values meaning that all affected working paths are eligible for restoration. The last column shows that the number of restored paths was the same as the number of paths restorable before capacity contention. This is because all contention was removed by mak-

ing all the routes in the DRSs node disjoint from one another. These results show that the solutions obtained in Table 6.16 are indeed 100% node failure restorable.

**TABLE 6.17. Minimum capacity single node failure restorable solutions: Node failure restoration using the Optimal Conflict Resolution method.**

<b>Trial</b>	<b>Transiting Affected Paths</b>	<b>Paths Restorable Before Capacity Contention</b>	<b>Optimally Restored Paths</b>
<b>1</b>	71	71 (100%)	71 (100%)
<b>2</b>	71	71 (100%)	71 (100%)
<b>3</b>	71	71 (100%)	71 (100%)

### 6.8.3 Overall Node Failure Restorability Results

The total capacities for 100% node failure restorable designs of the architectures considered in this study are found in Table 6.18. Three types of  $p$ -cycle solutions are presented. The first one, yielding the lowest total capacity of all the architectures is the flow  $p$ -cycle solution. Recall that flow  $p$ -cycles [42][43] are the most general type of  $p$ -cycle where cycles placed in the network provide protection to path segments. These segments may be entire end-to-end working paths or single spans. FIPP  $p$ -cycles as well as span-protecting  $p$ -cycles are simply specialized instances of flow  $p$ -cycles. It is not surprising that this architecture achieved the lowest total capacity. The downside of this architecture, however, is that operationally, failure localization is required before restoration can take place. Furthermore, the complexity of the protection structures makes this architecture undesirable in practice. Another  $p$ -cycle architecture result provided in Table 6.18, aside from the FIPP  $p$ -cycle result, is the result obtained using a modified node encircling  $p$ -cycle (NEPC) method. The original purpose of NEPCs [76] was to provide span-protecting  $p$ -cycles with the capability to recover from node failures. By definition, NEPCs must cross every node one hop away from the node they are intended to protect. The modified method relaxes this constraint, and also adding a constraint that only small flows are to be protected as part of this architecture. Small flows, in this context are defined as two span path seg-

ments. The node adjacent to both of the spans in this path segment is the node that node failure restorability is provided to. The idea behind small flows is to reduce the operational complexity of this architecture. It basically attempts to provide protection to the smallest possible path segment that allows for node failure restorability. In this case, the recovery action is only taken by the end-nodes of the small flows, where the small flows interface with their protection cycle. Solutions obtained using the small flow NEPC yield the most total capacity usage out of all the  $p$ -cycle architectures.

The DSP solution had to be modified very slightly to upgrade the span restorable design to a node restorable one. Only 4 additional units of capacity were required, corresponding to less than a 1% capacity increase relative to the minimum capacity single span failure solution. Finally, the PXT solution provided in Table 6.18 was obtained using a DRS method similar to the method used for FIPP  $p$ -cycles. Specifically, it was made sure that the DRSs generated contained working paths that were mutually node disjoint. Also, it was made sure the PXTs eligible to protect these DRSs provided protection paths that were node disjoint from the working paths they were assigned to protect. Overall, this PXT method yielded the solutions with the highest capacity use out of all the architectures.

**TABLE 6.18. Total capacities of 100% node restorable minimum capacity solutions for each architecture under study.**

Architecture	Total Capacity
Flow $p$ -Cycles	341
<b>FIPP <math>p</math>-Cycles (DRS method, best trial)</b>	383
<b>Relaxed NEPC with small flows</b>	439
DSP	449
<b>PXTs (DRS method, best trial)</b>	454

## 6.9 Conclusion

This chapter contained a series of studies that focused on the design and analysis of FIPP  $p$ -cycle networks in the context of optical transport networking. Specifically areas of optical transport networking that were addressed were generation of single span failure restorable minimum capacity/cost designs, dual-failure restorability, wavelength assignment, transparent reach and single node failure analysis. Because of the equipment assumptions made, the  $z$ -case path/cycle relationship had to be addressed in order to make sure that FIPP  $p$ -cycle designs were 100% single-failure restorable. Approaches were developed for determining dual-failure restorability as well as nodal R1 for minimum capacity 100% single span failure restorable designs. Additionally, wavelength assignment methods as well as a method to extract transparent reach data from designs were developed. The results obtained for FIPP  $p$ -cycles were contrasted against other pre-connected architectures under study.

Overall the minimum capacity solutions generated by FIPP  $p$ -cycles were efficient; they ranked 2nd in total capacity use to optimal span-protecting  $p$ -cycle designs. The main source of efficiency for FIPP  $p$ -cycle solutions came from the use of large, highly loaded DRSs protected by very large cycles. This effect was also observed in the JCP single span failure restorable minimum capacity solution (generated to meet the 20 maximum wavelength constraint) as well as minimum capacity single node failure restorable design. The large size of the cycles and the corresponding large protected DRSs were favourable in terms of minimum wavelength assignment because there is a relationship between the number of configurations in the design and the upper bound on the minimum number of wavelengths required. On the other hand, the lower bound on wavelength usage is related to the capacity on the maximally loaded span. Interestingly, it was observed that because minimum capacity routing was used and because the network graph provided very few options for cycle formation, the large cycles were forced to cross the maximally loaded spans that were crossed by the working paths they were protecting. In this case the constraint that the protection structures had to be cycles required that spare capacity be reserved on the

maximally loaded spans for the cycle despite the fact that this capacity may not have actually been reserved to protect any demands. This pushed the total capacity on the maximally loaded span up and increased the minimum number of wavelengths needed to implement the design.

While being great for capacity efficiency, the use of large DRS/cycle configurations played a significant role in explaining why FIPP  $p$ -cycles also yielded lower results in terms of dual-failure restorability. The large cycles ended up providing very long protection paths which were often much longer than the working paths they were protecting. Enabling these protection paths seized large quantities of spare capacity after the occurrence of a single failure. This seizure of spare capacity (sometimes affecting nearly the entire cycle) left all working paths that were also relying on this spare capacity without any means of restoration should a second failure affect them. Also, a second failure affecting any of the spans crossed by the protection path enabled after the first failure also resulted in unrestored service. The long protection paths also played a factor in transparent reach analysis. A method similar to the one used for  $z$ -case removal was used to prevent a large portion (in the order of 45% for initial minimum capacity designs) of the protection paths from exceeding the maximum transmission distance.

In general, there was observed a relationship between capacity use and the elegance with which a network design deals with situations that it was not originally optimized for. In a way, the point of view here is that additional capacity in the network is a kind of an investment, or insurance policy against unexpected occurrences or accidents. As more spare capacity is invested, less spare capacity sharing is required between different working paths for 100% failure restorability, resulting in shorter dedicated protection paths and thus an improvement in both the transmission reach and dual-failure restorability.



## 7.1 Introduction

The following chapter summarizes the research done as part of a three year (November 2005 - November 2008) collaboration between TRILabs and Nokia Siemens Networks under the name of High-Availability Network Architectures (HAVANA). The primary focus of the collaboration was the investigation and comparison, in the context of optical transport networking, of several pre-connected network survivability architectures. The main participants in this study include the author (FIPP  $p$ -cycles), Dr. Aden Grue (PXTs), Brian Forst (DSP and APS), Diane Onguetou (span-protecting  $p$ -Cycles), as well as Dr. Wayne D. Grover, Dr. Matthieu Clouqueur and Dr. Dominic Schupke. This chapter summarizes the third and final year of the study from the perspective of the FIPP  $p$ -cycle architecture. The overall goal of this chapter is the investigation of how the architectures under study could be implemented in a real transport network and how they would compare in terms of capital expenditure needed. The methods and results of this chapter were published in part in [67].

In this chapter we first introduce the NOBEL cost model [77] that is used to implement the architectures and provide relevant assumptions as well as equipment costs. Following this, the minimum capacity solutions used for the study are introduced and, as an initial exercise, the NOBEL model is applied to these solutions. Next, a new approach called FIPP maximum unit path straddler iterative heuristic (FIPP MUPS IH) for generation of minimum capacity FIPP  $p$ -cycle solutions, where only a single path per straddling route is protected, is introduced. This new approach is motivated by opportunities for cost reduction discovered in the initial costing exercise. Finally, the NOBEL cost model is then applied to the new solutions obtained using the FIPP MUPS IH method. The results are presented and contrasted against results gathered for other architectures under study.

The key concept on which the research in this chapter is based on is that of end-to-end optical transparency. In optical networking this means allowing transient traffic to remain in the optical domain end-to-end. This is done by optically cross-connecting the transiting paths at the intermediate nodes so that the payload does not need to be brought out of the optical domain. Upon failure, the entire lightpath is redirected onto a pre-determined and optically pre-cross-connected protection path. The NOBEL cost model provides equipment costs for transparent, opaque as well as hybrid architectures. As will be shown, there is strong cost incentive for moving away from opaque and towards transparent networking for all path-protecting architectures under study. This is reflected by the choices made when generating network designs intended for the implementation exercise. In particular, the use of regenerators is avoided by ensuring that every path, working and backup, in the designs does not exceed the maximum transmission distance allowed by the cost model. Additionally, wavelength conversion at transiting nodes is also avoided in path-protecting architectures by asserting that every shared protection structure as well as every path end-to-end uses only a single wavelength. The concept of same wavelength protection, where both the working path and protection path use the same wavelength, is shown to be one of the strongest forces in reducing the equipment cost in this study. Before further discussion of this concept can take place, the following section introduces the NOBEL cost model in detail.

## **7.2 NOBEL Cost Model**

The NOBEL cost model was developed, with contributions from major network suppliers and operators, as part of the European NOBEL project [78]. This model provides normalized costs for various relevant network elements and nodal equipment at the WDM layer. The intent of the model was to enable studies comparing different network alternatives as well as to provide a consistent reference for future

research in optical transport networks. In particular, this study was aimed at comparing the costs associated with transparent over traditional opaque networking.

The cost model for the study is divided into three classes. The first class, described in detail in Section 7.2.1, covers the reach and capacity independent nodal equipment. The second class, described in Section 7.2.2, covers the equipment needed to realize the working and protection paths on spans. The third class, covered in Section 7.2.3, covers the transmission equipment. This class is heavily dependant on the transmission distance; a metric that is directly influenced by the working paths and backup paths provided by protection structures as well as whether same wavelength protection is used or not.

The cost model is based on the currently prevalent data rate of 10 Gbits/s. All network costs include the basic infrastructure needed to support the network equipment such as power supplies, racks and software. Costs refer to complete bidirectional network elements. The cost model covers equipment that uses 40 and 80 wavelength channels. For this study, only 40 wavelength equipment was used. This choice is validated later in this chapter by making sure that the designs do not exceed this wavelength limit. This band of wavelengths is implemented using a single fiber per span (bidirectionality is assumed).

To calculate the transmission distance, or transmission reach, of a path, equation (6.10) from Chapter 6 is reused in this section. For completeness, this equation is given again in (7.1) where  $R$  is the total transmission distance,  $h$  is the number of hops of the path being considered,  $i$  is the insertion loss and  $L$  is the sum of the geographical lengths of the spans crossed by the path. Similarly to Chapter 6, the insertion loss is 80km for every hop taken by the path. Three levels of maximum transmission distances (MTDs) for network equipment have been defined. These distances are 750km, 1500km, 3000km. The equipment level used is determined by the transparent reach of the working or protection path being considered. Equipment

with the lowest MTD is used whenever possible. For example, paths with a transparent reaches of 740km and 760km will be implemented using equipment with MTDs of 750km and 1500km, respectively.

$$R = (h - 1) \cdot i + L \quad (7.1)$$

### 7.2.1 Nodal Equipment

The nodal equipment as proposed by the NOBEL model consists of optical switching devices such as reconfigurable optical add/drop multiplexers (ROADMs) and optical cross-connects (OXC) as well transparent node amplifiers (TDAs). ROADMs and OXC are similar in function to one another, the only difference being a ROADMs are used whenever there are 2 input fiber ports, and OXC are used whenever the number of input fiber ports is between 3 and 10 inclusive. The costs of ROADMs and OXC include multiplexing/demultiplexing costs as well as optical supervisory costs and optical power control. The cost of amplification modules is given separately as part of the TDA cost. TDAs are used at each bidirectional fiber pair on the node and are designed to compensate for the 80km insertion loss of the transparent node. ROADM, OXC and TDA costs are given using equations (7.2), (7.3) and (7.4), respectively, where  $N$  is the number of input fiber ports.

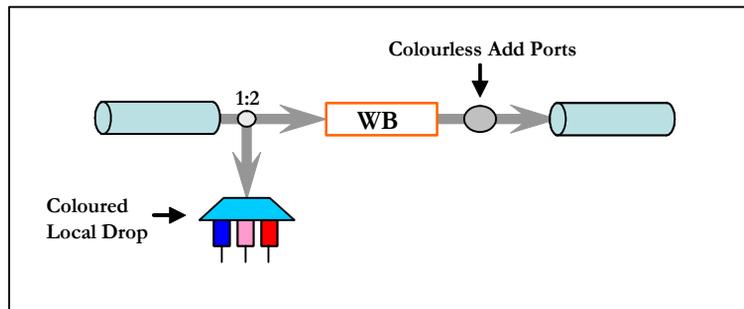
$$\text{ROADM Cost} = 11.8 \quad N = 2 \quad (7.2)$$

$$\text{OXC Cost} = \begin{cases} 5.35N + 2 & 3 \leq N \leq 5 \\ 5.85N + 2 & 6 \leq N \leq 10 \end{cases} \quad (7.3)$$

$$\text{TDA Cost} = 1.25N \quad (7.4)$$

The transparent node architecture proposed in the NOBEL cost model used to implement the transparent ROADM is shown in Figure 7.1. This architecture is based on wavelength blocking (WB) devices. These devices prevent the wavelengths that are to be locally dropped at the node from traversing through. Additionally, these

devices also incorporate wavelength power equalization for the channels that pass through the node. Using the same equalization capability, the power level of the channels that are to be blocked can be strongly reduced. The signal coming into the ROADM is duplicated using a passive 1:2 splitter for local drop. The second version of the signal goes through the WB where the dropped channels are suppressed. Finally colourless add ports, realized by a passive star coupler, provide the add functionality to the ROADM.



**FIGURE 7.1. Nodal structure of a 2 input fiber port WB based transparent ROADM. Unidirectional schematic shown. Adapted from [77].**

The schematic for the transparent OXC proposed in the NOBEL cost model is provided in Figure 7.2. This transparent node architecture is based on wavelength selective switches (WSS) in a broadcast and select (B&S) structure. WSSs are similar in operation to WBs in that they are able to suppress the wavelength channels that are to be locally dropped and are able to equalize the power levels of transiting channels. The main difference is that a WSS provides multiple input ports and is able to pick out individual channels from the input WDM combs and send them to the output port. This capability allows for devices that are based on WSSs, such as the OXC proposed here, to scale linearly as the number of fiber ports to the device increases. The add/drop functionality in the OXC is provided using the coloured local drop and colourless add ports, similarly to the transparent ROADM. As part of the B&S architecture, a 1:(N-1) coupler is used after the 1:2 splitter to broadcast the complete spec-

trum to the WSSs where it is decided which channels are to be suppressed and which are to be passed through.

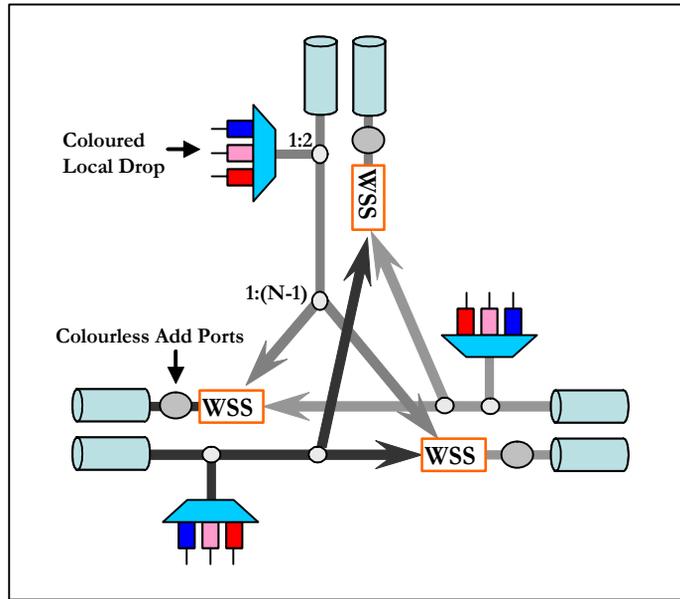


FIGURE 7.2. Nodal structure of a WSS based transparent OXC. Unidirectional schematic shown. Adapted from [77].

### 7.2.2 Span Equipment

The span equipment consists of dispersion compensating fiber (DCF) modules, as well as inline amplifiers (IAs) and dynamic gain equalizers (DGEs). In order to extend transmission distance, IAs and DCFs are used to compensate for attenuation and dispersion of the fiber, respectively. DCF modules and IAs are installed at the beginning of every 80km segment of a span. DGEs help extend the transmission distance by balancing the signal-to-noise ratios of different optical channels and by compensating for nonlinearities of optical amplifiers. A DGE is required at every fourth amplifier site. It is assumed that the equipment level used on a span depends on the path with the longest transmission distance crossing that span. For example if a span is crossed by three paths with transmission distances of 100km, 1400km and 1750km,

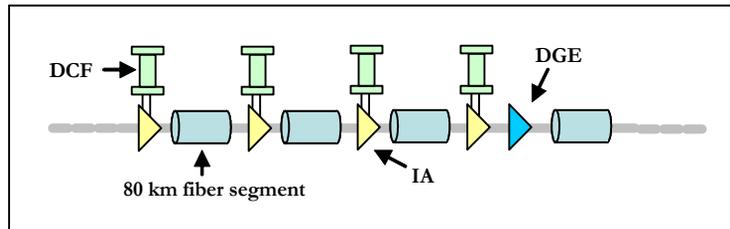
equipment with MTD of 3000km will be used on that span to accommodate the path with the longest transmission distance. The DCF, IA and DGE costs per span are given in equations (7.5), (7.6) and (7.7), respectively where  $L$  is the length of the span in kilometers and  $c$  is unit module cost based on the MTD of the equipment used.

$$\text{DCF Cost} = c \cdot \frac{L}{80[\text{km}]} \quad \text{where: } c = \begin{cases} 0.9 & \text{MTD} = 750\text{km} \\ 1.0 & \text{MTD} = 1500\text{km} \\ 1.2 & \text{MTD} = 3000\text{km} \end{cases} \quad (7.5)$$

$$\text{IA Cost} = c \cdot \left\lceil \frac{L}{80[\text{km}]} \right\rceil \quad \text{where } c = \begin{cases} 3.0 & \text{MTD} = 750\text{km} \\ 3.8 & \text{MTD} = 1500\text{km} \\ 4.7 & \text{MTD} = 3000\text{km} \end{cases} \quad (7.6)$$

$$\text{DGE Cost} = 3.0 \cdot \left\lceil \left\lceil \frac{L}{80[\text{km}]} \right\rceil \right\rceil \quad (7.7)$$

To summarize the network elements introduced in this section, an example of a unidirectional span segment is shown in Figure 7.3. From left to right, this illustration includes IAs and DCFs at the beginning of every 80km segment, with a DGE added at the fourth IA site. Span equipment costs provided in this study are for bidirectional network elements and the locations of the span network elements is reversed when the backwards direction is considered.



**FIGURE 7.3. Span equipment implementation. Unidirectional schematic shown.**

### 7.2.3 Transmission Equipment

The transmission equipment consists of reach dependant equipment such as tunable transponder (XPDR) cards, as well as electrical cross-connect (EXC) ports. These devices correspond to how a path interfaces with the transport network at the end-nodes. Transponders are devices that transmit an outgoing optical signal on the frequency that the transponder is tuned to. On the receiving end, transponders recover the payload form the optical carrier. EXCs can be used for add/drop traffic and perform electronic switching where signals cannot be switched optically. The cost of the EXC scales with the number of ports needed. The unit costs per XPDR for three MTD levels are given in (7.8). The EXC switch cost per 10G equivalent port is given in equation (7.9).

$$\text{XPDR Card Cost} = \begin{cases} 1.0 & \text{MTD} = 750\text{km} \\ 1.4 & \text{MTD} = 1500\text{km} \\ 1.9 & \text{MTD} = 3000\text{km} \end{cases} \quad (7.8)$$

$$\text{EXC Port Cost} = 0.28 \quad (7.9)$$

### 7.2.4 Generalized Node Model

The generalized node model based on the hybrid transparent node architecture proposed by the NOBEL cost model is illustrated in Figure 7.4. This model incorporates electrical add/drop ports as wells as optical add/drop ports for non-electrically switched paths. The EXC interfaces with the ROADM/OXC using electrical ports connected to coloured transponder cards. TDAs are also included as part of this model for completeness despite not being counted as part of the transmission cost. Combined, Figure 7.3 and Figure 7.4, provide a visual summary for the placement of all the network elements considered in this study.

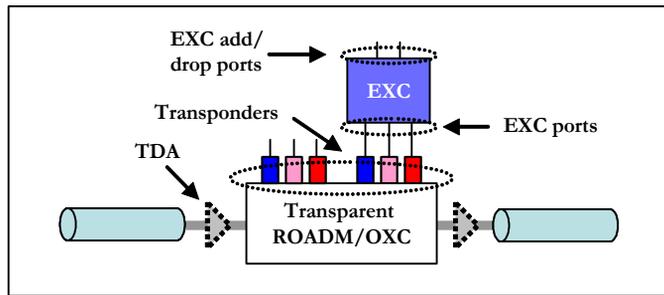


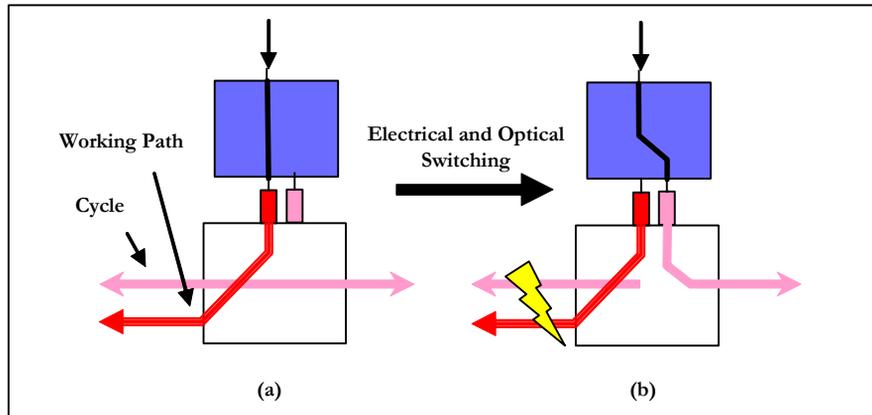
FIGURE 7.4. Generalized node model. Bidirectional schematic shown.

### 7.3 FIPP $p$ -Cycle Implementation

Where the previous section covered the NOBEL cost model in the most general case, this section addresses how FIPP  $p$ -cycles, specifically, are implemented using the proposed network equipment. The node costs covered in the previous section are dependant on the number of input ports coming into the node. FIPP  $p$ -cycles are implemented using a single fiber per span (the reverse direction is assumed as well) carrying 40 wavelengths. The number of input ports to any node in this case is equivalent to the nodal degree. The span costs covered in the previous section are calculated based on the longest path crossing that span. This can be calculated after all the transmission distances of all working and backup paths and the exact spans these paths cross are determined. Most importantly, the node and span costs are independent of the type of relationship that the working path has with the cycle, be it on-cycle or straddling and can be calculated in the same way for all architectures.

The transmission equipment cost, on the other hand, is highly dependant on the type of relationship that the working path has with the cycle protecting it. Before we can examine how straddlers and on-cycle paths can be implemented, it is important to introduce two distinct end-node implementations based on the generalized node model provided in the previous section. These two implementations correspond

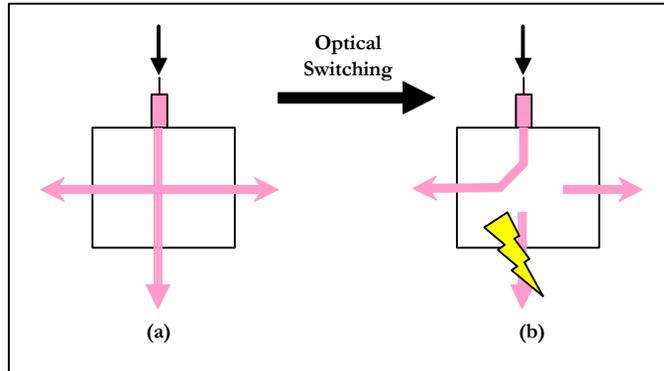
to whether a working path is protected using same wavelength protection or not. The first implementation, illustrated in Figure 7.5, covers the scenario where the working path is protected using different wavelength protection. This is the case where the working path and the cycle providing the protection path for it are not implemented using the same wavelength. For this implementation, 2 XPDR cards are required: one for the working path, and one to break into the cycle when the working path fails. The XPDRs must both be of high enough MTD level to support the working and protection path, respectively. Additionally, 3 EXC ports are required: one add/drop port and two ports to interface the EXC with the XPDR cards. Both, electrical and optical switching is required in this case. The figure only shows a single node, and this structure needs to be duplicated at the other end-node to complete the path. Note that the colours of the thick arrowed lines and the small vertical rectangles represent the wavelengths used by these paths and XPDR cards, respectively.



**FIGURE 7.5. End-node implementation of a working path under different wavelength protection. (a) Pre-failure state. (b) Failure state.**

The second end-node implementation, illustrated in Figure 7.6, covers the scenario where the working path is protected using same wavelength protection. Both the working path and the cycle providing the protection path to it use the same wavelength in this case. This implementation only requires one XPDR per end-node of the

working path as long as the XPDR used is of a high enough MTD level to support the longer of the two paths. In case of working path failure, only optical switching is required. This implementation is much more cost efficient than the implementation using different wavelength protection. How these implementations relate to the type of relationship that the working path has with the protection cycle, be it on-cycle or straddler, is covered in the following subsections.



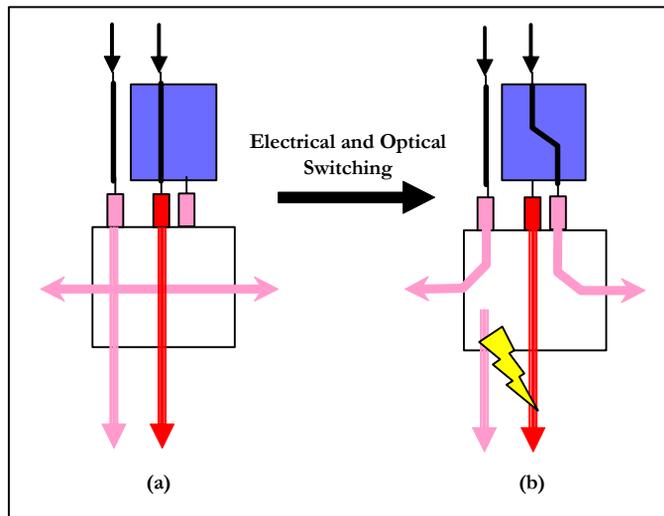
**FIGURE 7.6. End-node implementation of a working path under same wavelength protection. (a) Pre-failure state. (b) Failure state.**

### 7.3.1 On-cycle Implementation

Recall that on-cycle working paths are paths that cross at least one span also crossed by the cycle assigned to protect them. The cycle and working path cannot use the same wavelength on the spans they share. Since both the cycle and working path end-to-end must each be implemented using a single wavelength, the protection path provided by the cycle to the working path must use a different wavelength than the working path. This means that every on-cycle working path must be implemented using different wavelength protection as shown in Figure 7.5.

### 7.3.2 Straddler Implementation

Recall that in a straddling relationship, two working paths crossing the same spans end-to-end can be protected by the same cycle as long as the cycle crosses the same end-nodes as the working paths but does not cross any spans crossed by the working paths. Of the two protected working paths, one can be protected using same wavelength protection shown in Figure 7.6. There are no wavelength conflicts between this path and the cycle because the two are mutually span disjoint. The second path protected by this cycle must use different wavelength protection if the first working path uses the same wavelength as the cycle as shown in Figure 7.5. The two working paths are not span disjoint and cannot use the same wavelength on those spans. The above description of a straddling route implementation where the working paths are protected using same and different wavelength protection is the lowest cost implementation possible for a scenario where two working paths are protected on the straddling route by the same cycle. This is illustrated in Figure 7.7. Note that if the straddling route has only a single working path that requires protection, this working path can be protected using same wavelength protection.



**FIGURE 7.7. Implementation of an end-node of a straddling working route where one of the working paths is protected using same wavelength protection. (a) Pre-failure state. (b) Failure state.**

## 7.4 New Minimum Hops 3k Limited Non-z-Case DRS Solutions

As an initial costing exercise, the minimum capacity solutions presented in Chapter 6 were to be implemented using the NOBEL cost model. However, the two eligible minimum capacity non-z-case FIPP  $p$ -cycle solutions from Chapter 6 could not be directly used. The solution without reach limit constraints has paths exceeding the 3000km transmission distance limit imposed by the NOBEL cost model. The reach limited non-z-case solution was generated with a maximum transmission distance of 2000km, which is too short for this study. Because of this, new minimum capacity solutions were generated that were 3000km reach limited and contained no z-cases. These solutions were obtained using the same method as the minimum capacity solutions from Chapter 6: the DRS method (originally from [40]). Additionally, the least hops working routing used in the non-z-case 2000km reach feasible solutions from Chapter 6 was reused in the new solutions.

For the new minimum capacity solutions, consistent with the approach used in Chapter 6, it is assumed that every working path is required to have one pre-determined backup path provided by the cycle protecting that path. This means that no working path in the new solutions is to be in a z-case relationship with the cycle. Recall that a z-case is a path-cycle relationship where the two possible protection paths provided by the cycle to the working path are not span disjoint from that working path. In the general case, FIPP  $p$ -cycle solutions allow for this cycle-path relationship to exist. It is known that only one of the provided protection paths can fail at the same time as the working path, leaving the second protection path intact. The remaining protection path can be used to recover the failed working path. The exact protection path that is used is only known after a failure occurs. Since this violates the assumption in this study that each working path is to have a single pre-defined protection path that can be switched to in case of failure, no working path in a z-case relationship can be protected.

The new solutions were made 3000km reach feasible using the method for obtaining reach feasible solutions outlined in Chapter 6. Briefly, any path where the MTD limit was exceeded was removed from the generated DRSs before the DRSs were passed to the solver. The same procedure was performed for any  $z$ -cases found. This made sure that the solver returned a solution that was both reach feasible and contained no  $z$ -cases. The parameters used to generate the new minimum hops 3000km reach limit feasible non- $z$ -case solutions are as follows: Every demand was represented in at least 15 DRSs with 10 cycles generated for every DRS and the DRS size was set to have a maximum of 6 working routes. Similarly to Chapter 6, three trials were run. These trials differed by the seed used for the randomization process used in DRS generation. For comparison purposes and as a continuation of the HAVANA project, the solutions were obtained using the Germany network and the same demand relations as used in Chapter 6. The DRS SCP FIPP  $p$ -cycle model was implemented using AMPL 10.1 and solved using CPLEX 10.1. Experiments were terminated with a mipgap of 0.01.

The minimum capacity solutions for  $p$ -cycles, DSP, APS and PXT solutions, introduced in Chapter 6, were reused for comparison purposes here. Briefly, optimal minimum capacity  $p$ -cycle solutions were generated using the ILP for optimal  $p$ -cycle network design originally found in [35]. To generate the  $p$ -cycle solutions, all possible simple cycles from the Germany network were enumerated and used as an input to the ILP. DSP solutions were generated using the method in [31]. APS solutions were generated using the same method, but by simply making sure that only two routes between the end-nodes of every demand are generated. Finally, PXTs were generated using the DRS method [40]. For this method the parameters were as follows: The number of DRSs that each demand appeared in was 30, 30 candidate trails were generated for each DRS and the maximum size of every DRS was 15 demands. These solutions were reach limited to 1500km in this study using the same approach used to arrive at 2000km solutions presented in Chapter 6. The approach was to simply

inspect the set of DRSs before this set was passed to the solver, and to remove any protection relationship that violated the reach limit.

## 7.5 Minimum Capacity Results

The total capacity as well as the number of cycles used in the minimum capacity non-z-case 3000km reach limit feasible solutions obtained using the DRS method are presented in Table 7.1. The total capacities shown are for each trial. How these compare with other architectures under study is shown in Table 7.2 where FIPP  $p$ -cycles are ranked second in capacity use. Recall that any cycle/DRS configuration can be implemented using at most two wavelengths. One wavelength for the cycle and for one of the two working paths on any straddling route. And a second wavelength that is different from the cycle wavelength can be used to implement on-cycle paths as well as straddling paths that are not the same wavelength as the cycle. Given that there are at most 20 cycles in any of the solutions obtained, it is safe to assume that at most 40 wavelengths will be needed to implement the minimum capacity solutions. This validates the assumption that only 40 wavelength nodal equipment proposed by the NOBEL model is to be used for these FIPP  $p$ -cycle designs.

**TABLE 7.1. FIPP  $p$ -cycle (DRS method) minimum capacity single span failure restorable designs: no z-case, 3000km reach feasible.**

	<b>Total Capacity</b>	<b>Cycles Used</b>	<b>Max. Number of Wavelengths Required</b>
<b>Trial 1</b>	389	20	40
<b>Trial 2</b>	373	18	36
<b>Trial 3</b>	367	17	34

TABLE 7.2. Total capacities of the architectures under study.

	Total Capacity
<i>p</i> -Cycles (opaque)	317
FIPP <i>p</i> -Cycles DRS method (ave.)	~376
PXTs (1500 km)	401
DSP (same WL)	445
APS (1:1 and 1+1)	453

As was shown in Chapter 6, solutions obtained using the DRS method are composed of a set of configurations the cycles of which sometimes provide more protection relationships than is required at no extra cost. An example of this is when a cycle protects a straddling route where only a single unit of demand requires protection. Additionally, since the DRS method simply chooses the best combination of predetermined DRSs and cycles, sometimes the solver will return configurations that provide more protection relationships than required for certain demands. This is done as long as picking these configurations improves the objective value. In these situations of over-protection it is not immediately clear which segment of which cycle is to be used to protect the over-protected demands. In order to overcome this ambiguity and to satisfy the criteria that every unit working path must have a single protection path that is automatically switched to in case of a failure as well as to be able to apply the NOBEL cost model, the method of assigning actual protection to over-protected demands from Chapter 6 was used. Working paths were assigned to cycles they straddle first. On-cycle relationships were considered for protection after no more straddlers remained. After every working path was assigned, the costs associated with implementing these solutions can be obtained.

## 7.6 Initial Costing Exercise Results

The equipment cost breakdowns for the three trials used in the initial costing exercise are shown in Table 7.3. All three trials were implemented using one fiber per

span using 40 wavelength equipment. Because of this, the node costs for all three trials are equivalent. Recall that span equipment costs correspond to the transmission distances of paths crossing each span. Upon closer inspection, every span from Trial 3 used equipment rated for MTD of 3000km. This means that the span cost of 335 is highest span cost possible for designs implemented using 40 wavelengths or less. Trial 1 and Trial 2 had one span each where the equipment used was rated for MTD of 1500km and 750km, respectively. The rest of the spans on both Trial 1 and 2 used 3000km equipment.

The transmission costs for each trial are also shown in Table 7.3. Recall that these costs are highly dependant on whether the working path can be protected using same wavelength protection or not. In an effort to implement the solutions using the lowest transmission cost, same wavelength protection was used whenever possible. For every straddling route that is protected, one of the paths on this route was protected using the same wavelength as the cycle. Same wavelength protection was also used if a straddling route only requires that only one working path is protected. On-cycle paths, as discussed earlier in Section 7.3.1 were implemented using different wavelength protection. Table 7.4 shows the actual number of straddling working paths that use the same wavelength as the cycle for each trial. It turns out that at most 22 out of 97 working paths are protected using same wavelength protection. This means that 75 of the remaining working paths are on a different wavelength than the cycle which explains the high transmission costs associated with implementing these solutions using the NOBEL cost model.

**TABLE 7.3. FIPP  $p$ -cycle (DRS method) equipment assignment results.**

	<b>Node Cost</b>	<b>Span Cost</b>	<b>Transmission Cost</b>	<b>Total Cost</b>
<b>Trial 1</b>	374	331	594	1299
<b>Trial 2</b>	374	328	589	1290
<b>Trial 3</b>	374	335	599	1308

**TABLE 7.4. Number of working paths (out of 97) using same wavelength protection in FIPP  $p$ -cycle (DRS method) solutions.**

	<b>Working paths using same wavelength protection</b>
<b>Trial 1</b>	19
<b>Trial 2</b>	22
<b>Trial 3</b>	20

Table 7.5 shows how the FIPP  $p$ -cycle solutions compare to the other architectures under study. The node costs are the same for all architectures that do not exceed 40 wavelengths, which only 1:1 APS exceeded. Because of this 1:1 APS was implemented using a second fiber on select spans. This additional fiber is reflected in the increased node cost where additional fiber ports had to be considered. It is also reflected in an increase in span costs where additional IAs, DGEs and DCFs were required on the newly added fiber.

The span costs for all architectures, not including 1:1 APS because of the additional fiber added, reflect the longest transmission distances crossing individual spans for each architecture. Span-protecting  $p$ -cycles yielded the lowest cost in this category, as they are implemented using wavelength conversion at every node for every cycle and working path, meaning that the longest path crossing every span has the transmission distance equal to the length of the span itself. The longest span in the Germany network is only 294km meaning that span equipment with MTD of 750km can be used on every span in the network for span-protecting  $p$ -cycles. This corresponds to the lowest span equipment cost possible for solutions not exceeding 40 wavelengths implemented using a single fiber per span where every span in the network requires span equipment. FIPP  $p$ -cycles perform poorly in this category as they use the highest MTD equipment on almost every single span. The reason for this is because the cycles used in minimum capacity solutions tend to be quite large. The result of this is that the backup paths provided by these cycles tend to be very long,

many exceeding the transmission distance of 1500km and forcing the use of equipment with MTD of 3000km on each span they cross.

The transmission costs of each architecture are also shown in Table 7.5. The transmission costs for FIPP  $p$ -cycles are very high because of the high number of working paths protected using different wavelength protection, as mentioned above. DSP and 1:1 APS have the lowest transmission costs. This is because for both architectures the working paths and backup paths are very short, as opposed to FIPP  $p$ -cycles where the number of demands protected by each cycle is maximized in an effort to minimize capacity. This results in very large cycles which in turn makes the lengths of the protection paths provided by these cycles very long. Additionally, both DSP and 1:1 APS use same wavelength protection for all 97 working paths in the network, meaning that 1 XPDR per end-node is used for both the working and the protection paths and no electrical switching is required at all. Note the difference between 1:1 and 1+1 APS in terms of transmission cost. 1:1 APS allows for the working path and the backup path to share 1 XPDR per end-node and optically switch to the protection path when the working path fails. 1+1 APS must transmit the same signal on the working and the backup path at the same time meaning that 2 XPDRs are needed per end-node, which explains why the transmission cost of the latter approximately doubles that of the former.

**TABLE 7.5. Equipment costs of the architectures under study.**

	<b>Node Cost</b>	<b>Span Cost</b>	<b>Transmission Cost</b>	<b>Total Cost</b>
<b>DSP (same WL)</b>	374	275	209	858
<b>PXTs (1500 km)</b>	374	275	289	937
<b>1:1 APS (same WL)</b>	436	313	207	956
<b>1+1 APS</b>	374	275	403	1052
<b><math>p</math>-Cycles (opaque)</b>	374	225	532	1131
<b>FIPP <math>p</math>-Cycles DRS method (average)</b>	374	~331	~594	~1299

Overall, the lowest total cost architectures are DSP, PXTs and 1:1 APS according to Table 7.5. FIPP  $p$ -cycles have the highest total cost relative to all other architectures. The biggest contributor to this high cost is the number of working paths not protected using same wavelength protection. Where at most 22 working paths are protected using same wavelength protection in FIPP  $p$ -cycle solutions, all 97 working paths are protected using same wavelength protection in the case of DSP, PXTs and 1:1 APS. Furthermore, because the cycles tend to be very long, the backup paths provided by these cycles are also long thus further contributing to the equipment cost. The approach in the following sections takes these observations into consideration and proposes a new ILP heuristic algorithm that aims to lower the equipment cost needed to implement the newly generated designs.

## 7.7 New Approach: Maximize Same Wavelength Protection

The initial costing exercise from the previous sections yielded the following insight: The main reason that the costs for implementing architectures such as DSP and 1:1 APS were the much lower than FIPP  $p$ -cycles was because these architectures were able to use same wavelength protection for all the working paths in the network where FIPP  $p$ -cycles were not. Recall that in FIPP  $p$ -cycles, two straddling paths crossing the same spans can be protected by the same cycle. One of these two paths can be protected using same wavelength protection, while the second path must be protected using a different wavelength than the cycle. The new approach, FIPP  $p$ -Cycle Maximum Unit Path Straddler Iterative Heuristic (FIPP MUPS IH), introduced in the following section attempts to maximize the number of working paths protected using same wavelength protection under the NOBEL cost model. The main goal of the new approach is to maximize the number of straddling relationships and only allow for each straddler to protect a single unit of demand. Note that this is different from the way that straddlers are conventionally treated, where they get two protection paths per cycle. Not allowing FIPP  $p$ -cycles to protect two units of demand per straddler

removes one of the core features of FIPP  $p$ -cycles that allow them to be very capacity efficient. In this study, however, it is not capacity that we are trying to minimize, but the cost of implementation.

By maximizing the total number of unit straddlers protected, this new approach directly affects the transmission cost in that the total number of EXC ports and XPDR cards is reduced when the NOBEL cost model is applied. The new approach also has the additional benefit of reducing the lengths of the backup paths used. This affects span costs as well as transmission costs as equipment of lower MTD levels can be used, relative to the solutions from the initial costing exercise. This is because a unit straddling path effectively cuts the cycle into two segments, either of which are eligible to be used as a protection path. Not only do these protection paths tend to be shorter than the protection paths provided to the cycle for on-cycle working paths but now that only a unit straddling path needs to be protected, the shortest of the two eligible protection paths can be used.

## 7.8 FIPP Maximum Unit Path Straddler Iterative Heuristic

The FIPP  $p$ -Cycle Maximize Unit Path Straddlers Iterative Heuristic (FIPP\_MUPS\_IH) algorithm is illustrated in Figure 7.8. New working routing is first generated using the *GenerateWorkingRouteMaxStraddlers* algorithm. This is done because inspection of initial exercise DRS based solutions, where least hops working routing was used, showed that some working routes crossed spans in a way that prevented a simple cycle from being formed that could protect this route as a straddler. It was observed, however, that if some of these demands could be re-routed, then cycles could be found to protect them as straddlers. The working route generation algorithm aims to route working capacity for as many demands as possible in a least hops way that guarantees that a cycle can be generated to protect these demands as straddlers. For demand relations whose working routing could not be protected in a

straddling relationship, least hops routing is generated such that a cycle exists that can protect this path on-cycle but without a z-case relationship. Additionally the working routes, whether on-cycle or straddling, are reach limited by the generation algorithm to ensure that they can be implemented using the NOBEL cost model.

Once the working routing is determined for every demand, the algorithm begins to generate cycles as well as sets of demands those cycles protect. Recall that a set of protected demands as well as a cycle protecting those demands is referred to as a configuration. Configurations are optimally generated one at a time using *GenerateMaxUnitProtStraddlerConfigILP*. This ILP returns a configuration that aims to protect the most demand relations using a cycle that traverses as few hops as possible. Additionally, the cycle returned is guaranteed to contain a reach limit feasible backup path for every demand protected by that cycle. It is also guaranteed that the backup paths are span disjoint from the working paths so that the solution is 100% single span failure restorable. Once the configuration is returned by the ILP, it is recorded as part of the solution. The demands protected by this configuration are removed from the set of remaining unprotected demands. If the set of unprotected demands is non-empty, the process returns to the beginning of the loop and the ILP is called again. The loop terminates when there are no more demands left to protect. The solution returned by the FIPP MUPS IH algorithm contains enough cycles to protect all the straddler and non-straddler demands in the network against single failures while also maximizing the total number of demands protected as a straddler. The solution does not contain any z-cases and cycles protecting straddlers only provide a single protection path to these straddlers.

```

FIPP_MUPS_IH {
    GenerateWorkingRoutesMaxStraddlers();

    repeat {
        solve GenerateMaxUnitProtStraddlerConfigILP;
        Record the new configuration;
        Remove the units demands protected by the new configuration
        from the set of remaining unprotected demands;
    } while the set of remaining unprotected demands is not empty;
}

```

FIGURE 7.8. The FIPP\_MUPS\_IH algorithm.

### 7.8.1 Working Route Generation

To find the new working routes for every demand relation, the *GenerateWorkingRoutesMaxStraddlers* algorithm depicted in Figure 7.9 was used. This algorithm generates working paths for every demand relation such that the total number of potential straddling relationships is maximized. For a given relation  $r$ , first the *GenerateMinHopsStraddlingRouteILP* is called. The goal of this ILP is to return a minimum hops working route that is guaranteed to have at least one cycle in the network that can protect this route as a straddler. This ILP also makes sure that the working route and its potential protection path are both within the transparent reach limit. Upon success, the demand relation is flagged as a demand that is to be protected as a straddler and the new working route is recorded. Whenever *GenerateMinHopsStraddlingRouteILP* results in infeasibility, meaning that no working route is found that could be protected as a straddler, *GenerateMinHopsOnCycleRouteILP* is called for that demand relation. The goal of this ILP is to return a least hops working route that is to be protected on-cycle, such that at least one cycle exists that can protect this route without there being a  $z$ -case.

```

GenerateWorkingRoutesMaxStraddlers {
  for every demand relation r do {
    solve GenerateMinHopsStraddlingRouteILP[r];
    if (success) then {
      Mark the demand to be protected as a straddler;
      Record the new working routing for this demand;
    } else if (failure) then {
      solve GenerateMinHopsOnCycleRouteILP[r];
      if (success) then {
        Mark the demand to be protected as a non-straddler;
        Record the new working routing for this demand;
      } else if (failure) then {
        Error: Cannot find two or more disjoint routes.
      }
    }
  }
}

```

**FIGURE 7.9.** **GenerateWorkingRoutesMaxStraddlers** algorithm. Used to generate working routes for every demand relation such that the total number of potential straddling relationships is maximized.

## 7.8.2 Generate Minimum Hops Straddling Route ILP

The *GenerateMinHopsStraddlingRouteILP* works by generating three mutually span disjoint least hops routes (*Route1*, *Route2* and *Route3*) for demand relation *r*. The ILP sorts these routes by hops if possible in ascending order with *Route1* containing the fewest hops. *Route1* corresponds to the least hops working path being generated, while *Route2* and *Route3* correspond to two segments of a least hops cycle that can protect *Route1*. *Route2*, as a result of the routes being sorted, should contain fewer hops than *Route3* and is intended to be viewed as a potential protection path for *Route1*. The ILP makes sure that both *Route1* and *Route2* are reach limit feasible. Note that the cycle formed by *Route2* and *Route3* is only a placeholder to make sure that the working path for demand relation *r* (*Route1*) is routed correctly. It is not used in the solution generated by FIPP MUPS

IH. The fact that all three routes are mutually span disjoint makes sure that *Route1* can be protected as a straddler. To make sure that the cycle formed by *Route2* and *Route3* is simple, these two routes must also be mutually node disjoint. Recall that a simple cycle is a cycle that does not cross any node more than once. *GenerateMinHopsStraddlingRouteILP* uses the following sets, parameters, variables and constraints.

**Sets:**

$S$  Set containing the spans of the network.

$N$  Set containing the nodes of the network.

$S^n$  Set of spans adjacent to node  $n$ .  $S^n \subseteq S, n \in N$ .

$D$  Set of demand relations.

$N^r$  Set of end-nodes for relation  $r$ .  $N^r \subseteq N, r \in D$ .

$R3$  Set containing the three disjoint routes to be generated.  
 $\{Route1, Route2, Route3\} \in R3$ . *Route1* corresponds to the working path being generated. *Route2* and *Route3* respectively correspond to two segments of a potential cycle eligible to protect *Route1*.

**Parameters:**

$M''$  A positive constant.  $M'' = 100$ .

$M'$  A positive constant.  $M' = 10$ .

$H$  Insertion loss penalty corresponding to power loss of the signal passing through nodal equipment.  $H = 80[km]$ .

$R$  Maximum transparent reach.

$c^s$  This parameter corresponds to the length of span  $s$  in  $[km]$ .  $s \in S$ .

### Variables:

$\mu^{s,q}$  Equal to 1 if span  $s$  is used by route  $q$  and is equal to 0 otherwise.  
 $s \in S, q \in R3$ .

$v^{n,q}$  Equal to 1 if node  $n$  is used by route  $q$  and is equal to 0 otherwise.  
 $n \in N, q \in R3$ .

### GenerateMinHopsStraddlingRouteILP:

$$\text{Minimize: } \sum_{s \in S} (M'' \cdot \mu^{s, Route1} + M' \cdot \mu^{s, Route2} + \mu^{s, Route3}) \quad (7.10)$$

### Constraints:

$$\sum_{s \in S^n} \mu^{s,q} = 1 \quad \forall n \in N^r, \forall q \in R3 \quad (7.11)$$

$$v^{n,q} \geq \mu^{s,q} \quad \forall n \in N, \forall s \in S^n, \forall q \in R3 \quad (7.12)$$

$$\sum_{s \in S^n} \mu^{s,q} = 2 \cdot v^{n,q} \quad \forall n \in N, n \notin N^r, \forall q \in R3 \quad (7.13)$$

$$\sum_{s \in S} c^s \cdot \mu^{s,q} + H \cdot \left( \sum_{n \in N} v^{n,q} - 1 \right) \leq R \quad \forall q \in R3, q \neq Route3 \quad (7.14)$$

$$v^{n, Route2} + v^{n, Route3} \leq 1 \quad \forall n \in N, n \notin N^r \quad (7.15)$$

$$\sum_{q \in R3} \mu^{s, q} \leq 1 \quad \forall s \in S \quad (7.16)$$

The objective function (7.10) minimizes the total capacity used by the three routes generated for the demand relation  $r$ . The objective function also makes sure that, of the three routes generated, the least hops route corresponds to *Route1*. Constraint (7.11) makes sure for all three routes being generated, exactly one span adjacent to the origin and the destination nodes of demand  $r$  is used. Constraint (7.12) ensures that, for any span used by one of the three routes, the nodes adjacent to that span are also marked as used by that route. Constraint (7.13) makes sure that the routes generated are continuous end-to-end. This is accomplished by making sure that every node used as part of a route (excluding the end-nodes of demand  $r$ ), must have exactly two used adjacent spans. Constraint (7.14) makes sure that *Route1* and *Route2*, corresponding to the demand's working and possible protection route respectively, do not exceed the transparent reach limit. Constraint (7.15) makes sure that *Route2* and *Route3* are node disjoint from one another. Lastly, constraint (7.16) is used to make sure that the three routes generated are mutually span disjoint.

### 7.8.3 Generate Minimum Hops On-Cycle Route ILP

*GenerateMinHopsOnCycleRouteILP* works mostly in the same way that *GenerateMinHopsStraddlingRouteILP* does. Three least hops routes (*Route1*, *Route2* and *Route3*) are generated and sorted in an ascending order where *Route1*, corresponding to the working path being generated, spans the fewest hops. Similarly to *GenerateMinHopsStraddlingRouteILP*, *Route2* and *Route3* represent the cycle segments making up a cycle that can protect *Route1*. *Route2*, span-

ning the fewer number of hops of the two segments, corresponds to a potential protection path for *Route1*. The ILP makes sure that both *Route1* and *Route2* are reach limit feasible.

The only main difference between *GenerateMinHopsStraddlingRouteILP* and *GenerateMinHopsOnCycleRouteILP* is that the latter only aims to make sure that the working path being generated can be protected on-cycle. This is reflected in the constraints where it is no longer necessary for all three paths to be mutually span disjoint. *Route1* and *Route2* have to be span disjoint so that the working path generated can be guaranteed a backup path that is span disjoint from it to protect against single span failures. *Route2* and *Route3* have to be span and node disjoint so that a simple cycle that can protect *Route1* can be formed. *Route1* and *Route3* no longer have to be span disjoint and can cross the same spans. In fact, *Route1* and *Route3* will never be span disjoint when *GenerateMinHopsOnCycleRouteILP* completes. This is because, as part of *GenerateWorkingRoutesMaxStraddlers* this ILP is only called if it is not possible to protect demand relation  $r$  as a straddler, which is determined by *GenerateMinHopsStraddlingRouteILP* resulting in infeasibility. The following new constraints are required.

**Constraints:**

$$\mu^{s, Route1} + \mu^{s, Route2} \leq 1 \quad \forall s \in S \quad (7.17)$$

$$\mu^{s, Route2} + \mu^{s, Route3} \leq 1 \quad \forall s \in S \quad (7.18)$$

*GenerateMinHopsOnCycleRouteILP* uses the objective function (7.10) and constraints (7.11), (7.12), (7.13), (7.14) and (7.15). Two new additional constraints (7.17) and (7.18) replace constraint (7.16). Where

*GenerateMinHopsStraddlingRouteILP* ensured that the three working routes generated were all mutually span disjoint, *GenerateMinHopsOnCycleRouteILP* only needs to make sure that *Route1* and *Route2* are span disjoint from one another (constraint (7.17)) and that *Route2* and *Route3* are also mutually span disjoint (constraint (7.18)).

## 7.8.4 Generate Maximum Unit Protection Straddler Configuration ILP

*GenerateMaxUnitProtStraddlerConfigILP* takes as an input the working routing for demand relations that still require protection and generates a configuration composed of a cycle and a set of demand relations protected by it. Every working route that can be protected as a straddler, as determined by the *GenerateWorkingRoutesMaxStraddlers* algorithm is protected as a straddler in the configuration being generated. Only one unit of demand is protected as part of a straddling route. The cycle generated for the configuration provides reach limited protection paths to the demands in the protected path set. In addition to the previously introduced sets, parameters and variables, the *GenerateMaxUnitProtStraddlerConfigILP* uses the following.

### Sets:

$N^s$  Set of nodes adjacent to span  $s$ .  $N^s \subseteq N, s \in S$ .

### Parameters:

$d3^r$  This parameter contains the number unit demands for relation  $r$  that are to be protected as straddlers.  $r \in D$ .

$d2^r$  This parameter contains the number unit demands for relation  $r$  that cannot be protected as straddlers and must be protected in an on-cycle relationship.  $r \in D$ .

$\beta^{u,r}$  Equal to 1 if node  $u$  is an end-node for relation  $r$  and is 0 otherwise.

$\pi^{s,r}$  Equal to 1 if the working route of demand  $r$  crosses span  $s$  and is 0 otherwise.  $s \in S, r \in D$ .

$\alpha$  This is the minimum amount that the potential values differ by.  
 $\alpha = |\mathcal{M}|^{-1}$ .

### Variables:

$\Lambda 3^r$  Equal to 1 if demand relation  $r$  is protected as part of the configuration being generated and is equal to 0 otherwise. This variable only takes into account demands that can be protected as straddlers in this configuration.  $r \in D$ .

$\Lambda 2^r$  Equal to 1 if demand relation  $r$  is protected as part of the configuration being generated and is equal to 0 otherwise. This variable only takes into account demands that cannot be protected as straddlers and must be protected in on-cycle relationships.  $r \in D$ .

$\lambda^u$  Equal to 1 if node  $u$  is an end-node for the demand protected by the configuration being generated and is 0 otherwise.  $u \in N$ .

$e^{(u,v)}$  Equal to 1 if span  $s$  adjacent to node  $u$  and node  $v$  is used as part of the cycle being generated and is 0 otherwise.

$$s \in S, u \in N^s, v \in N^s, u \neq v.$$

$z^u$  Equal to 1 if node  $u$  is crossed by the cycle being generated, and is equal to 0 otherwise.  $u \in N$ .

$k^u$  Equal to 1 if node  $u$  is the root node and is equal to 0 otherwise.  $u \in N$ .

$p^u$  Non-integer variable that stores that potential value of node  $u$ .  $0 \leq p^u \leq 1, u \in N$ .

$\Omega^{s,r}$  Equal to 1 if the backup path for relation  $r$  crosses span  $s$  and is 0 otherwise.  $s \in S, r \in D$ .

$\omega^{u,r}$  Equal to 1 if the backup path for relation  $r$  crosses node  $u$  and is 0 otherwise.  $u \in N, r \in D$ .

### GenerateMaxUnitProtStraddlerConfigILP:

$$\text{Maximize: } \left[ \sum_{s \in S, r \in D} M \cdot \pi^{s,r} \cdot (\Lambda 3^r + \Lambda 2^r) \right] - \left[ \sum_{s \in S, u \in N^s, v \in N^s, u \neq v} e^{(u,v)} \right] \quad (7.19)$$

### Demand Protection Constraints:

$$\Lambda 3^r \leq d 3^r \quad \forall r \in D \quad (7.20)$$

$$\lambda^u \geq \Lambda 3^r \cdot \beta^{u,r} \quad \forall u \in N^r, \forall r \in D \quad (7.21)$$

$$\Lambda 2^r \leq d 2^r \quad \forall r \in D \quad (7.22)$$

$$\lambda^u \geq \Lambda 2^r \cdot \beta^{u,r} \quad \forall u \in N^r, \forall r \in D \quad (7.23)$$

### Cycle Generation Constraints:

$$e^{(u,v)} + e^{(v,u)} \leq 1 \quad \forall s \in S, \forall u \in N^s, \forall v \in N^s, u \neq v \quad (7.24)$$

$$\sum_{s \in S^u, v \in N^s, u \neq v} e^{(u,v)} + e^{(v,u)} = 2 \cdot z^u \quad \forall u \in N \quad (7.25)$$

$$\sum_{u \in N} k^u \leq 1 \quad (7.26)$$

$$\sum_{s \in S^u, v \in N^s, u \neq v} e^{(u,v)} \leq 1 + k^u \quad \forall u \in N \quad (7.27)$$

$$p^v - p^u \geq \alpha \cdot e^{(u,v)} - (1 - e^{(u,v)}) \quad \forall s \in S, \forall u \in N^s, \forall v \in N^s, u \neq v \quad (7.28)$$

$$z^u \geq \lambda^u \quad \forall u \in N \quad (7.29)$$

### Disjointness Constraints:

$$\sum_{r \in D} (\pi^{s,r} \cdot \Lambda 3^r) + e^{(u,v)} \leq 1 \quad \forall s \in S, \forall u \in N^s, \forall v \in N^s, u \neq v \quad (7.30)$$

$$\sum_{r \in D} \pi^{s,r} \cdot (\Lambda 3^r + \Lambda 2^r) \leq 1 \quad \forall s \in S \quad (7.31)$$

### Backup Path Generation Constraints:

$$\omega^{u,r} = \beta^{u,r} \cdot (\Lambda 3^r + \Lambda 2^r) \quad \forall u \in N^r, \forall r \in D \quad (7.32)$$

$$\sum_{s \in S^u} \Omega^{s,r} = \omega^{u,r} \quad \forall u \in N^r, \forall r \in D \quad (7.33)$$

$$\omega^{u,r} \geq \Omega^{s,r} \quad \forall s \in S^u, \forall u \in N, \forall r \in D \quad (7.34)$$

$$\sum_{s \in S^u} \Omega^{s,r} = 2 \cdot \omega^{u,r} \quad \forall u \in N, u \notin N^r, \forall r \in D \quad (7.35)$$

$$\Omega^{s,r} \leq e^{(u,v)} + e^{(v,u)} \quad \forall s \in S, \forall u \in N^s, \forall v \in N^s, u \neq v, \forall r \in D \quad (7.36)$$

### Backup Path Reach Limit and z-Case Avoiding Constraints:

$$\sum_{s \in S} c^s \cdot \Omega^{s,r} + H \cdot \left( \sum_{u \in N} \omega^{u,r} - 1 \right) \leq R \quad \forall r \in D \quad (7.37)$$

$$\Omega^{s,r} + \pi^{s,r} \leq 1 \quad \forall s \in S, \forall r \in D \quad (7.38)$$

The objective function (7.19) has the following format:

$$\text{Maximize}(M \cdot \text{TotalCapacityOfDemandsProtected} - \text{CycleCapacity})$$

The main goal is to maximize the number of demands protected by the configuration being generated while at the same time minimizing the capacity used by the cycle protecting said demands. A constant  $M$  is applied to *TotalCapacityOfDemandsProtected* to make sure that a meaningful solution is returned even when the total capacity used by the cycle is greater than the total capacity used by the protected demands. Constraints (7.20) and (7.21) account for demands that are to be protected as straddlers. Constraint (7.20) makes sure that only non-zero demands are eligible for protection. Constraint (7.21) makes sure that the end-nodes of any demand that is protected as part of the configuration are marked for protection as well. These nodes are used as part of the cycle generation section of the model. In order for a demand to be protected by a cycle, the cycle must cross the end-nodes of that demand. Constraints (7.22) and (7.23) work in the same way as (7.20) and (7.21) but account for the demands that must be protected as non-straddlers.

Constraints (7.24) to (7.29) govern cycle generation. These are based on an ILP for generating span-protecting  $p$ -cycles found in [79]. This method works by designating one of the nodes crossed by the cycle as the root node. The rest of the cycle is effectively grown from the root node using unidirectional vectors. Two vectors are placed such that their head nodes are at the root node. The rest of the vectors are aligned head to tail such that two directed paths are made, sourced by the root node. Where these two paths meet, and the only place where the tail nodes of two vectors coincide, is referred to as a reversal node. It is these vectors that mark the spans crossed by the cycle being generated. Potential values are assigned to each node crossed by the cycle. These potential values are arranged in such a way that for any vector, the head node of the vector must have a lower potential value than the tail node. The function of these values is to make sure that only one cycle is formed. Without these constraints, the ILP would generate multiple cycles whenever doing so results in the fewest units of capacity used; something that is undesirable when the goal is to make sure that a single continuous backup path exists for every demand protected. This is a problem also encountered in the classic traveling salesman problem (TSP). The problem of making sure that a single tour is generated instead of several subtours is referred to as subtour elimination.

Constraint (7.24) makes sure that at most one vector crosses any one span or, in other words, that the cycle never crosses the same span more than once. Exactly two spans adjacent to a node crossed by the cycle must be crossed by cycle vectors by constraint (7.25). Constraints (7.26) and (7.27) make sure that at most one root node exists and that this node sources exactly two vectors, respectively. Constraint (7.28) sets the potential at the head node of any vector to be less than the potential at the tail of the vector. Finally (7.29) makes sure that the nodes crossed by the cycle include the end-nodes of the protected demands.

Constraints (7.30) and (7.31) govern the relationships between the cycle and the straddling and non-straddling demands. Constraint (7.30) makes sure that all pro-

tected straddling demands are span disjoint from the protection cycle and from other protected straddling demands. Constraint (7.31) makes sure that all protected non-straddling demands are span disjoint from all other protected demands.

Constraints (7.32) to (7.36) generate a single backup path for every protected demand. The technique used for generating backup paths is the same as the technique used to generate working routes in *GenerateMinHopsStraddlingRouteILP* and *GenerateMinHopsOnCycleRouteILP*. The only additional constraint is that the backup path must cross the same spans as the cycle. Constraint (7.32) makes sure that the end-nodes of the demand and the backup path are the same. Constraint (7.33) ensures that the end-nodes of the backup path cross exactly one adjacent span. Constraint (7.34) makes sure that every node adjacent to any span crossed by the backup path is also crossed by the backup path. Constraint (7.35) makes sure that any backup path crossing a node that is not an end-node of that path, must also cross two of that node's adjacent spans. The backup path generated must cross the same spans as the cycle by constraint (7.36).

Lastly, to make sure that the solution can be implemented using the NOBEL cost model, constraint (7.37) makes sure that the backup path generated does not exceed the transparent reach limit. The final constraint (7.38) prevents z-case relationships. It does so by making sure that any protected working path and its protection path are span disjoint. If no backup path is found that is within the transparent reach limit and is disjoint from the working path, the demand for which this protection path was generated cannot be added to the set of demands protected by the cycle.

## **7.9 Minimum WL Assignment for Unit Path Straddler Solutions**

In order to verify that the solutions generated using the FIPP MUPS IH method can be implemented using 40 wavelength equipment as part of the NOBEL

cost model, a minimum wavelength assignment model was used. This model is a modified version of the *MinWavelengthAssignmentILP* from Section 6.5. It was modified to reflect the fact that straddling routes could now only support a single protected working path and that this path is required to use the same wavelength as the cycle protecting it. The resulting model called *MinWavelengthAssignmentUPSILP* is given below. This model reuses the sets, parameters, variables given in Section 6.5 as well as the new set and constraint introduced below.

**Sets:**

$D^{p''}$  Set of demand relations that are in a straddling relationship with cycle  $p$ .  $p \in P, D^{p''} \subseteq D$ .

**Constraints:**

$$w^{\lambda, p, c} = w^{\lambda, r, c} \quad \forall \lambda \in WL, \forall p \in P, \forall r \in D^{p''}, \forall c \in (C^r \cap C^p) \quad (7.39)$$

The new model uses the objective function (6.1) and constraints (6.2) through (6.6) from Section 6.5. An additional constraint (7.39) is added to make sure that every straddling path is assigned the same wavelength as the cycle protecting it.

## 7.10 Experimental Setup

Two types of minimum capacity solutions were generated using the FIPP MUPS IH method: One having a reach limit of 3000km and the other having a reach limit of 1500km. This was done by setting parameter  $R$  for every ILP called as part of the FIPP MUPS IH method to 3000 and 1500, respectively. The solutions were obtained using the Germany network and using the same demand matrix as was used in the minimum capacity DRS solutions as well as for all other architectures under study, originally introduced in Chapter 6. The FIPP MUPS IH algorithm was imple-

mented using AMPL 10.1 and solved using CPLEX 10.1. Every ILP called as part of the FIPP MUPS IH method was terminated with a mipgap of 0.0001.

## 7.11 Results

The capacity breakdown for the solutions obtained using the FIPP MUPS IH method are shown in Table 7.6. Additionally the capacity breakdown of the solutions obtained using the DRS method are shown as well, averaged over the three trials. The working capacity of the MUPS solutions increased by close to 20 units relative to the DRS solutions. This was the result of the re-routing that took place so that demands could be protected as straddlers where they could only be protected on-cycle before, when strictly least hops routing was used. The spare capacity also increased, as expected for MUPS solutions, relative to the DRS solutions because it was no longer possible to protect two units of demand per straddler. This is further reflected in the high number of cycles used in the MUPS solutions. Recall from Table 7.2 that FIPP DRS solutions were the second best in terms of capacity use. The highest capacity use was from APS, coming in at 453 units. The new MUPS solutions use more capacity than this, making these new solutions the highest, in terms of capacity use.

**TABLE 7.6. FIPP  $p$ -cycle minimum capacity single span failure restorable, non- $z$ -case, reach limit feasible designs: FIPP DRS and FIPP MUPS IH solutions.**

	<b>Working Capacity</b>	<b>Spare Capacity</b>	<b>Total Capacity</b>	<b>Cycles Used</b>
<b>FIPP MUPS IH Solution (1500km)</b>	185	360	545	34
<b>FIPP MUPS IH Solution (3000km)</b>	184	350	532	29
<b>FIPP DRS Solution (average)</b>	166	-210	-376	-18.3

Between the two MUPS solutions presented in Table 7.6, the one reach limited to 1500km used more capacity as well as cycles than the 3000km solution. This was expected because the 1500km solution was constrained on the size of cycles that it may use and was forced to place several cycles where the 3000km solution could use

just one. Table 7.7 provides a closer look at the configuration types generated for the two MUPS solutions. The two configuration types presented are all straddler configurations as well as on-cycle/straddler configurations. The table shows approximately no difference between the 3000km and the 1500km solution in terms of all straddler configurations. The 1500km solution had 6 more straddler/on-cycle configurations, reflecting that generally on-cycle working paths have longer protection paths than straddling paths. Where certain on-cycle paths could be protected by the same cycle in the 3000km solution, these paths had to be protected by smaller different cycles in the 1500km solution.

The minimum number of wavelengths required, obtained by *MinWLAssignmentUPSILP*, is also presented in Table 7.7. Note that the 1500km solution can actually be implemented using fewer wavelengths than the 3000km solution. This is because, while more cycles were used in the 1500km solution, these cycles were smaller and used fewer same wavelength channels over the spans on the network. This gave more opportunities for wavelengths assigned to cycles and the unit straddlers protected by them to be reused by on-cycle working paths protected by other cycles, resulting in fewer wavelengths needed overall. Ultimately, the fact that fewer than 40 wavelengths are needed in both cases validates the assumption that 40 wavelength equipment can be used when the NOBEL cost model is applied.

**TABLE 7.7. FIPP MUPS IH method solution configuration breakdown by types of paths protected and the minimum number of wavelengths required.**

	All Straddler Configurations	On-cycle/Straddler Configurations	Min. Number of Wavelengths Required
FIPP MUPS IH Solution (1500km)	18	16	34
FIPP MUPS IH Solution (3000km)	19	10	35

The results from applying the NOBEL cost model to the MUPS solutions are presented in Table 7.8. For comparison reasons, DRS solution equipment costs are also presented, averaged over the three trials. The node cost is the same for all solu-

tions presented because they were all implemented using one fiber per span using 40 wavelength equipment.

In terms of span costs, there was an improvement for both MUPS solutions compared to the DRS solutions. Recall that the DRS solution span costs corresponded to equipment with MTD of 3000km being used on almost every span in the network. The span cost corresponding to every span using 3000km equipment is 335 units. Also, recall from Section 7.6 that the lowest span cost possible, where every span in the network uses equipment with MTD of 750km, is 225 units, which was the case for span protection  $p$ -cycles. The MUPS 3000km solution has a span cost that is close to the 335 unit maximum limit. Upon, closer inspection of the MUPS 3000km solution all but 5 (out of 26) spans used 3000km equipment. Closer inspection of the MUPS 1500km solution showed that only one span used 750km equipment, with the rest of the spans having 1500km equipment installed. This corresponds to the data presented in Table 7.8 where the span cost of the MUPS 1500km solution falls approximately in the middle of the upper (335) and lower (225) span cost limits.

**TABLE 7.8. Equipment assignment results: Cost breakdown for FIPP MUPS IH and FIPP DRS solutions.**

	<b>Node Cost</b>	<b>Span Cost</b>	<b>Transmission Cost</b>	<b>Total Cost</b>
<b>FIPP MUPS IH Solution (1500km)</b>	374	273	350	997
<b>FIPP MUPS IH Solution (3000km)</b>	374	317	384	1075
<b>FIPP DRS Solution (average)</b>	374	-331	-594	-1299

The largest improvement, when comparing the DRS and MUPS solutions, is in terms of transmission cost, as shown in Table 7.8. Recall that every working path under same wavelength protection requires 1 XPDR card per end-node, while working paths protected using different wavelength protection require 2 XPDR cards plus an additional 3 EXC ports per en-node. Where at most only 22 (out of 97) working paths were protected using same wavelength protection as part of the DRS solutions, a

total of 65 (out of 97) working paths were protected using same wavelength protection in the MUPS solutions, as shown in Table 7.9. This is reflected in the improvement of 210 and 244 cost units when comparing the DRS solutions with the MUPS 3000km and 1500km solutions, respectively. Despite the fact that the main goal of the MUPS IH method was to maximize the number of straddlers protected in the solution, 32 units of demand were not protected as straddlers, and therefore could not be under same wavelength protection. The reason for this was because at least one end-node of these demands fell on a degree 2 node. Any cycle that protects a demand originating at a degree 2 node must cross both of the spans adjacent to the node. One of these spans must also be crossed by the protected working path so the cycle and the working path can never be mutually span disjoint. This means that the working path must be protected using different wavelength protection.

**TABLE 7.9. Number of working paths (out of 97) using same wavelength protection in FIPP MUPS IH solutions.**

	<b>Working paths using same wavelength protection</b>
<b>FIPP MUPS IH Solution (1500km)</b>	65
<b>FIPP MUPS IH Solution (3000km)</b>	65

Table 7.10 shows a revised version of Table 7.5, now also containing the costs related to the MUPS solutions along side of the costs associated the other architectures under study. Note that the lowest cost FIPP  $p$ -cycle solution obtained using the MUPS IH method with the reach limit set to 1500km, while being a great improvement over the DRS solutions, is still higher than costs associated with implementing DSP, PXTs as well as 1:1 APS. The main reason for this was because FIPP was only able to protect 65 out of 97 demands under same wavelength protection as discussed above, while DSP, PXTs and 1:1 APS were able to protect all 97 demands using same wavelength protection. Because of the 32 demands using different wavelength protection, the FIPP solutions were forced to use an additional 64 XPDR cards and 192 EXC ports. If it had been possible for FIPP MUPS IH to generate a solution where

every single path was protected as a straddler, that is if a network was used where every node was of degree 3 or higher, the costs associated with implementing FIPP  $p$ -cycle solutions would be much closer to the costs associated with 1:1 APS and PXTs.

**TABLE 7.10. Equipment costs of the architectures under study, revised.**

	<b>Node Cost</b>	<b>Span Cost</b>	<b>Transmission Cost</b>	<b>Total Cost</b>
<b>DSP (same WL)</b>	374	275	209	858
<b>PXTs (1500 km)</b>	374	275	289	937
<b>1:1 APS (same WL)</b>	436	313	207	956
<b>FIPP MUPS IH Solution (1500km)</b>	374	273	350	997
<b>1+1 APS</b>	374	275	403	1052
<b>FIPP MUPS IH Solution (3000km)</b>	374	317	384	1075
<b><math>p</math>-Cycles (opaque)</b>	374	225	532	1131
<b>FIPP DRS Solution (average)</b>	374	-331	-594	-1299

In addition to the discussion above, some general observations about FIPP MUPS IH solutions can be made. First, since only a single unit of demand is protected per straddler, the cycle provides two eligible protection paths for that working path. The solutions generated above only used the shortest of the two cycle segments as a protection path. However, it would be possible to enable dual-failure protection for these straddlers with very little additional cost as long as the longer of the two segments (to be used as the second protection path) is within the maximum reach limit and the XPDR card used per end-node of this working path is of a high enough MTD level to transmit over the longest of the three paths (one working and two protection). Furthermore, the MUPS solutions left some headroom in terms of how closely the architecture came to the 40 wavelength limit. For example, the MUPS 1500km solution can be implemented using at least 34 wavelengths. This means that additional demands may still be added and cycles for their protection formed without having to upgrade the 40 wavelength equipment already in the network. The costs associated

with adding the new demands to the network would be contained mostly in transmission costs, where a total of 1 XPDR cards per end-node would have to be added in the case of a straddler and 2 XPDR cards and 3 EXC ports would be needed per end-node for an on-cycle demand.

## 7.12 Conclusion

In this chapter, the costs associated with implementing FIPP  $p$ -cycle as well as other architectures using the NOBEL cost model were investigated. As an initial exercise, FIPP DRS solutions were implemented using the cost model. It was observed that same wavelength protection was much more cost effective than different wavelength protection in terms of capital expenditure. A new MUPS IH method was introduced with the aim of reducing the cost of implementation by maximizing the total number of unit demand protecting straddlers, since these are the only type of working path that can be protected using same wavelength protection. Two MUPS IH solutions were generated: one reach limited by 3000km the other reach limited by 1500km. The latter proved to be the strongest FIPP  $p$ -cycle solution in terms of cost. While a considerable improvement over DRS solutions was observed, the lowest possible costs obtained for architectures such as DSP, PXT and 1:1 APS could not be reached by FIPP  $p$ -cycle solutions. This is because the degree 2 nodes in the network prevented almost a third of the demands from being protected using same wavelength protection.

# Chapter 8: Conclusion

---

## 8.1 Main Contributions

In this thesis, the main intent was to explore the recently introduced FIPP  $p$ -cycle architecture; to provide new methods for generating FIPP  $p$ -cycle designs as well as to explore the architecture in the context of relevant transport networking topics so to compare it to other network survivability schemes previously presented in literature.

Four new design methods were introduced for FIPP  $p$ -cycle network design: FIPP Column Generation (CG), FIPP Iterative Heuristic (IH), FIPP Joint Capacity Placement (JCP) DRS ILP and finally FIPP Maximum Unit Path Straddler Iterative Heuristic (MUPS IH). Using the FIPP CG method, based on an OR technique called column generation, it was possible to generate optimal or near optimal FIPP  $p$ -cycle solutions that showed an improvement of as much as 36% in terms of cost, compared to the commonly used DRS method. This method considered a more general configuration type than what was possible with other methods in literature. Using the FIPP CG method it was possible to generate solutions that exhibited redundancies of 50%. This result is particularly efficient, compared to the redundancies of 100% or more found in currently deployed ring and APS systems. The FIPP IH method was introduced in order to generate feasible solutions that could be used as a starting point for the FIPP CG method. FIPP IH solutions were found to be comparable in capacity to the DRS results obtained. The results also showed that the FIPP IH method was able to generate solutions for network sizes larger than what was possible using the more commonly used FIPP DRS method.

The FIPP JCP DRS ILP method was developed in order to take advantage of the capacity saving opportunities presented by allowing working paths to be deviated from the shortest or least capacity routing. The new JCP DRS method was shown to

reduce the total capacity used by 13% relative to the original SCP DRS method, where only shortest working routing was permitted. It was observed that the most benefit gained from jointness came from adding just a single routing option per demand over the single shortest route per demand relation used in SCP. We found that allowing 3 or more working route options did not provide significant improvements. The additional working route options allowed for unit cycles to protect more than just two working paths per demand relation in cases where the routing options were mutually disjoint. This gave the solver additional configuration options to add to the final design. These options were not possible in SCP, under shortest routing only.

Lastly, the FIPP MUPS method was introduced to generate FIPP  $p$ -cycle designs where the number straddlers carrying only one unit of working capacity was maximized. This was done in order to utilize same wavelength protection; a cost effective method for protecting individual working paths under the NOBEL project cost model. It was found that the NOBEL cost model implementations of the MUPS designs yielded a significant reduction in cost (in the order of 23%) relative to the cost required for the implementation of the initial minimum capacity solutions obtained using the FIPP DRS method. The number of working paths in the new solutions that utilized same wavelength protection was 65 out of 97. The fact that the FIPP MUPS IH method could not get all of the working paths to use same wavelength protection was because 32 out of 97 working paths had at least one degree 2 node as an end-node. The cycles passing through these nodes were forced to cross both of spans adjacent to them, forcing any working path terminating at these nodes to be routed using a different wavelength than the cycle.

During the comparative studies performed as part of the HAVANA project it was found that FIPP  $p$ -cycles yielded 100% single span and 100% single node failure restorable designs that were the most efficient in terms of spare capacity use, relative to the other path-protecting architectures under study. It was found that the minimum capacity single span restorable designs provided node failure restorability to 84% of

the working paths for no extra cost and only 13.2% more spare capacity was required to achieve 100% single node failure restorability. As part of the HAVANA project, the minimum capacity single span failure restorable solutions were subjected to dual span failures to see how well they perform. It was found that over all possible dual-failure scenarios, only 53% of the working paths were restored. This was the lowest dual-failure restorability out of all the path-protecting architectures under study. Upon closer inspection, it was found that the minimum capacity solutions used very large cycles and DRSs, which impacted negatively on dual-failure restorability. The same minimum capacity solutions were analyzed in the context of transparent optical networking where the maximum transmission distance of 2000 km was imposed on every working and protection path in the design. This was done to simulate the attenuation and dispersion effects that would be encountered in real transparent optical networks. It turned out that 45.4% of the protection paths exceeded the 2000 km reach limit. These results were again attributed to the fact that very large cycles and DRSs were used in the minimum capacity solutions and their use forced the protection paths to be exceptionally long. It was found that an additional 22% more space capacity was required in order to achieve minimum capacity single span failure restorable solutions that were also 2000 km reach limit feasible.

Overall, the biggest insights gained as part of this thesis work was the fact that FIPP  $p$ -cycles performed the best (in terms of capacity use) on highly connected networks with a dense demand matrix. The reason for this was that these kinds of networks allowed the solver to generate highly loaded configurations containing many straddler relationships. It was found that in lower connectivity networks (containing degree 2 nodes, for example) FIPP  $p$ -cycles could not utilize straddling relationships whenever any working path originated or terminated at such a node. It was also found that sparser demand patterns forced the creation of configurations where large cycles were placed in order to protect relatively few working paths. Another interesting effect observed in FIPP designs was that extra spare capacity had to sometimes be provi-

sioned due to the fact that every FIPP protection structure was required to be a cycle. This capacity was provisioned even if there was no protection path that would utilize it and this effect was more common in network problems using sparse demand patterns.

The main benefit of FIPP  $p$ -cycles is that it is an architecture that is able specialize very well when designed for a specific metric in mind (such as minimizing capacity while protecting against single span failures). This specialization can negatively affect the performance of FIPP designs when subject to failure scenarios that were not designed for. The fact that this architecture can be so efficient actually gives it quite a bit of flexibility when it comes to designing for metrics such as dual-failure or node-failure restorability. FIPP  $p$ -cycles is still an architecture that was only very recently introduced and much still remains to be discovered about it. The properties of pre-connection and fast restoration speed, high capacity efficiency, path protection, failure independence, end-node recovery-action activation and low operational complexity are all properties that are attractive to network operators and are exhibited by FIPP  $p$ -cycles. Because of this, FIPP  $p$ -cycles are likely to remain a strong contender in the field of transport networking and will continue to generate interest for years to come.

## 8.2 Publications

Included below is the list of peer-reviewed published (or accepted for publication) works produced over the course of this thesis.

- B. Jaumard, C. Rocha, D. Baloukov, W. D. Grover "A Column Generation Approach for Design of Networks using Path-Protecting  $p$ -Cycles," Proceedings of the 6th International Workshop on Design of Reliable Communication Networks (DRCN 2007), La Rochelle, France, 7-10 October 2007.

- D. Baloukov, W. D. Grover, and A. Kodian, "Toward jointly optimized design of failure-independent path-protecting  $p$ -cycle networks," J. Opt. Networking, 7, 62-79 (2008)
- A. Grue, W. D. Grover, M. Clouqueur, D. Schupke, J. Doucette, B. Forst, D. Onguetou, D. Baloukov, "Comparative Study of Fully Pre-Cross-Connected Protection Architectures for Transparent Optical Networks," Proceedings of the 6th International Workshop on Design of Reliable Communication Networks (DRCN 2007), La Rochelle, France, 7-10 October 2007.
- A. Grue, W. D. Grover, M. Clouqueur, D. Schupke, D. Baloukov, D. Onguetou, B. Forst, "CAPEX Costs of Lightly Loaded Restorable Networks Under a Consistent WDM Layer Cost Model," to appear in the proceedings of IEEE International Conference on Communications (ICC 2009), Dresden, Germany, June 14-18, 2009.
- D. Onguetou, D. Baloukov, W. D. Grover, "Near-Optimal FIPP  $p$ -Cycle Network Designs using General Path-Protecting  $p$ -Cycles and Combined GA-ILP Methods," to appear in the Proceedings of the 7th International Workshop on the Design of Reliable Communication Networks (DRCN 2009), Washington, D.C., October 25-28, 2009.

### 8.3 Technical Reports/Presentations

Included below are the two major technical reports and three yearly presentations were generated for the HAVANA project.

- A. Grue, B. Forst, D. Onguetou, D. Baloukov, J. Doucette, A. Kodian, W. D. Grover, "Comparative Study of Fully Pre-cross-connected Protection Architectures for High Availability Transparent

Optical Networking: Summary Report on Progress November 2005 - October 2006", Trlabs Tech. Rep. ST-06-01, Nov. 9, 2006.

- A. Grue, B. Forst, D. Onguetou, D. Baloukov, W. D. Grover, "Comparative Study of Fully Pre-cross-connected Protection Architectures for High Availability Transparent Optical Networking: Project Year 2 Report", Trlabs Tech. Rep. ST-07-01, Nov. 30, 2007.
- A. Grue, B. Forst, D. Onguetou, D. Baloukov, J. Doucette, A. Kodian, W. D. Grover, First-year Slide Decks Produced for Nokia Siemens Networks, Project HAVANA, 2006.
- A. Grue, B. Forst, D. Onguetou, D. Baloukov, W. D. Grover, Second-year Slide Decks Produced for Nokia Siemens Networks, Project HAVANA, 2007.
- A. Grue, B. Forst, D. Onguetou, D. Baloukov, W. D. Grover, Third-year Slide Decks Produced for Nokia Siemens Networks, Project HAVANA, 2008.

## 8.4 Topics for Further Research

The method of column generation used in the design of FIPP  $p$ -cycle networks in Chapter 4 can be improved in several ways. First, it may be possible to set up the decomposition of the LP in a way that prevents the pricing problem from having a set of sub-tour elimination constraints that gets exponentially larger as the size of the network increases. Another approach is to keep the decomposition the same and use a pricing heuristic to generate the next column at the end of each iteration. Another area of the CG method to address is the initial starting solution. How close this initial solution is to optimality makes a huge difference in the runtime of the column generation method. The IH method introduced in this thesis scaled poorly with network

size and if a different ILP or a fast heuristic method for generating good solutions could be developed, the CG method then can be used to quickly tighten those solution to optimality.

Another area where further research can be done is on the MUPS method introduced in Chapter 7. Under the NOBEL cost model, it became apparent that in optical networks considerable cost reduction could be obtained when same wavelength protection, provided by the use of unit working path straddlers, was utilized. A full ILP of the MUPS method can be formulated to better utilize the resources needed to implement the resulting design. Additional transparent reach constraints can be added to the ILP to keep down the MTDs of the protection paths. This formulation would be closer to optimizing for the cost of the network directly. This formulation can also be extended to take modularity into consideration. This would allow for better modeling of the effects observed in real networks where the cost of the network does not come from paying for each individual wavelength channel. Instead the channels are paid for in sets and new channels are needed only when the current set is exhausted.

Another area to consider is the extension of the FIPP  $p$ -cycle designs to a multiple quality of protection (QoP) framework [80]. In extending this framework, each demand would be eligible for a variety of protection classes such as single span failure protection, dual span failure protection, node failure protection or no protection at all. Both SCP and JCP problems can be formulated under the QoP framework. It would allow for the investigation of the sensitivity of FIPP  $p$ -cycle designs to the different protection classes and it would be particularly interesting to see what the solutions look like when taking the different classes into consideration simultaneously.



## Chapter 9: References

---

- [1] A. J. Vernon, J. D. Portier, "Protection of Optical Channels in All-Optical Networks," Proceedings of the 18th Annual National Fiber Optic Engineers Conference (NFOEC 2002), pp. 1695-1706, Dallas, TX, September 2002.
- [2] H. Timmons, "Ruptures call safety of Internet cables into question," International Herald Tribune, Accessed March 18th, 2009, available online: <http://www.ihf.com/articles/2008/02/04/technology/cables.php>, February 4, 2008.
- [3] J. Borland, "Analyzing the Internet Collapse," ABC News, accessed March 18th, 2009, available online: <http://abcnews.go.com/Technology/story?id=4244474>, 5 February 2008.
- [4] R. Singel, "Cable Cut Fever Grips the Web," Wired, accessed March 18th, 2009, available online: <http://blog.wired.com/27bstroke6/2008/02/who-cut-the-cab.html>, 6 February 2008.
- [5] T. Philip, "Soon, undersea cable cut won't snap your Net," Economic Times, accessed March 18th, 2009, available online: [http://economictimes.indiatimes.com/Infotech/Internet/Soon\\_undersea\\_cable\\_cut\\_wont\\_snap\\_your\\_Net/articleshow/3910913.cms](http://economictimes.indiatimes.com/Infotech/Internet/Soon_undersea_cable_cut_wont_snap_your_Net/articleshow/3910913.cms), 30 December 2008.
- [6] G. Wearden, "Cable failure hits U.K. Net traffic," CNet News, accessed March 18th, 2009, available online: [http://news.cnet.com/2100-1037\\_3-5111964.html](http://news.cnet.com/2100-1037_3-5111964.html), 26 November 2008.
- [7] M. Reardon, "Sprint Nextel suffers service outage," CNet News, accessed March 19th, 2009, available online: [http://news.cnet.com/Sprint-Nextel-suffers-service-outage/2100-1037\\_3-6024922.html?tag=mncol](http://news.cnet.com/Sprint-Nextel-suffers-service-outage/2100-1037_3-6024922.html?tag=mncol), January 9, 2006.
- [8] R. Magley, "Melted cable caused massive outage," The Hub, accessed March 19th, 2009, available online: [http://www.ouraynews.com/Articles-i-2008-04-09-176841.112113\\_Melted\\_cable\\_caused\\_massive\\_outage.html](http://www.ouraynews.com/Articles-i-2008-04-09-176841.112113_Melted_cable_caused_massive_outage.html), April 9, 2008.
- [9] W. D. Grover, Mesh-Based Survivable Networks. Upper Saddle River, New Jersey, USA: Prentice Hall PTR, 2003.
- [10] A. Betker, C. Gerlach, R. Hülsermann, M. Jäger, M. Barry, S. Bodamer, J. Späth, C. Gauger, M. Köhn, "Reference transport network scenarios," technical report, BMBF MultiTeraNet, July 2003. [http://www.ikr.uni-stuttgart.de/IKRSimLib/Usage/Referenz\\_Netze\\_v14\\_full.pdf](http://www.ikr.uni-stuttgart.de/IKRSimLib/Usage/Referenz_Netze_v14_full.pdf).
- [11] P. Batchelor, B. Daino, P. Heinzmann C. Weinert, J. Späth, B. Van Caenegem, D.R. Hjelme, R. Inkret, H.A. Jäger, M. Joindot, A. Kuchar, E. Le Coquil, P. Leuthold, G. de Marchis, F. Matera, B.

- Mikac, H.P. Nolting, F. Tillerot and N. Wauters, "Ultra high capacity optical transmission networks: Final report of action COST 239," Faculty of Electrical Engineering and Computing, University of Zagreb, 1999.
- [12] D. E. Comer, *Internetworking with TCP/IP: Principles, Protocols, and Architectures*, 4th edition, Prentice Hall, 2000.
- [13] B. Forouzan, *Introduction to Data Communications and Networking*, Boston, Massachusetts, McGraw-Hill, 1998.
- [14] U. Black and S. Waters, *SONET and T1: Architectures for Digital Transport Networks*, 2nd edition, Upper Saddle River, New Jersey, Prentice Hall, 2002.
- [15] S. V. Kartalopoulos, *Introduction to DWDM Technology: Data in a Rainbow*, IEEE Press, Piscataway, NJ, 2000.
- [16] B. Davie and Y. Rekhter, *MPLS: Technology and Applications*, Morgan Kaufmann, San Francisco, California, 2000.
- [17] P. Tomsu, C. Schmutzer, *Next Generation Optical Networks: The Convergence of IP Intelligence and Optical Technologies*, Prentice Hall, Upper Saddle River, NJ, 2002.
- [18] ANSI, *Synchronous Optical Network (SONET) – Automatic Protection Switching*, ANSI T1.105.01-1995.
- [19] ANSI, *Synchronous Optical Network (SONET) – Basic Description including Multiplex Structure, Rates and Formats*, ANSI T1.105-1995, 1995.
- [20] ITU-T *Network Node Interfaces for the Synchronous Digital Hierarchy (SDH)*, ITU-T Recommendation G.707, 1993, 1996
- [21] R. Bhandari, *Survivable Networks: Algorithms for Diverse Routing*, Kluwer Academic Publishers, November 1998.
- [22] Bellcore, *SONET Dual-Fed Unidirectional Path Switched Ring (UPSR) Equipment Generic Criteria*, GR-1400-Core, Issue 1, Revision 1, October 1995.
- [23] Bellcore, *SONET Bidirectional Line-Switched Ring Equipment Generic Criteria*, GR-1230-Core, Issue 2, November 1995.
- [24] P. Neusy, R. Habel, "Availability Analysis of Optical Shared Protection Rings for Long-haul Networks," *Proceedings of Optical Fiber Communication Conference (OFC 1999)*, San Diego, CA, pp. TuL5-1–TuL5-4, February 1999.
- [25] M. W. Maeda, "Management and Control of Transparent Optical Networks," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 16, no. 7, pp. 1005-1023, September 1998.
- [26] M. Herzberg, S. J. Bye, A. Utano, "The Hop-Limit Approach for Spare-Capacity Assignment in Survivable Networks," *IEEE/ACM*

- Transactions on Networking, vol. 3, no. 6, pp. 775-784, December 1995.
- [27] R.R. Iraschko, M. MacGregor, W.D. Grover, "Optimal capacity placement for path restoration in STM or ATM mesh survivable networks," IEEE/ACM Transactions on Networking, vol.6, no.3, June 1998, pp. 325-336.
- [28] Y. Xiong, L. G. Mason, "Restoration strategies and spare capacity requirements in self-healing ATM networks," IEEE/ACM Transactions on Networking, vol. 7, no. 1, pp. 98-110, February 1999.
- [29] S. Kini, M. Kodialam, T. V. Laksham, S. Sengupta, C. Villamizar, "Shared backup Label Switched Path restoration," IETF Internet Draft, draft-kini-restoration-shared-backup-01.txt, work in progress, May 2001.
- [30] A. M. C. A. Koster, A. Zymolka, M. Jager, R. Hulsermann, "Demand-wise Shared Protection for Meshed Optical Networks," Journal of Network and Systems Management, vol. 13, no. 1, pp. 35-55, March 2005.
- [31] Brian Forst, Wayne Grover, "Factors Affecting the Efficiency of Demand-wise Shared Protection," Proceedings of the 6th International Workshop on Design of Reliable Communication Networks (DRCN 2007), La Rochelle, France, 7-10 October 2007.
- [32] A. Koster et al, "Demand-wise Shared Protection for Meshed Optical Networks," Proc. Design of Reliable Communication Networks, Banff, Alberta, Canada, Oct 19-22, 2003. pp. 85-92.
- [33] T.Y.Chow, F.Chudak, A.M. Ffrench, "Fast Optical Layer Mesh Protection Using Pre-Cross-Connected Trails," IEEE/ACM Transactions on Networking, Vol 12, No. 3, June 2004, pp 539-547.
- [34] S. Kim, S. Lumeta, "Capacity-Efficient Protection with Fast Recovery in Optically Transparent Mesh Networks," Proceedings of the First International Conference on Broadband Networks (BROADNETS 2004) , San José, California, USA, 25-29 October 2004.
- [35] W. D. Grover, D. Stamatelakis, "Cycle-oriented distributed preconfiguration: Ring-like speed with mesh-like capacity for self-planning network restoration," Proc. IEEE International Conference on Communications (ICC) '98, Atlanta, Georgia, USA, 7-11 Jun. 1998, pp. 537-543.
- [36] D. Stamatelakis, W. D. Grover, "Theoretical underpinnings for the efficiency of restorable networks using preconfigured cycles ( $p$ -cycles)," IEEE Transactions on Communications, vol. 48, no. 8, Aug. 2000, pp. 1262-1265.
- [37] W. Grover, D Stamatelakis, "Bridging the ring-mesh dichotomy with  $p$ -cycles", Proc. IEEE / VDE DRCN 2000, Munich, Germany, April

- 2000, pp. 92-104.
- [38] W. D. Grover, A. Kodiana, "Failure-Independent Path Protection with  $p$ -Cycles: Efficient, Fast, and Simple Protection for Transparent Optical Networks," Proceedings of the 7th International Conference on Transparent Optical Networks (ICTON 2005), Barcelona, Catalonia, Spain, pp. 363-369, July 3-7, 2005.
- [39] A. Kodian and W. D. Grover, "Failure-Independent Path-Protecting  $p$ -Cycles: Efficient and Simple Fully Pre-connected Optical Path Protection," J. of Lightwave Technology, Vol. 23, No. 10, pp. 3241-3259 (October 2005).
- [40] A. Kodian, W. D. Grover, J. Doucette, "A Disjoint Route Sets Approach to Design of Failure-Independent Path-Protecting  $p$ -Cycle Networks," Proceedings of the 5th International Workshop on Design of Reliable Communication Networks (DRCN 2005), Island of Ischia (Naples), Italy, pp. 231-238, 16-19 October 2005.
- [41] D. Onguetou, D. Baloukov, W. D. Grover, "Near-Optimal FIPP  $p$ -Cycle Network Designs using General Path-Protecting  $p$ -Cycles and Combined GA-ILP Methods," to appear in the Proceedings of the 7th International Workshop on the Design of Reliable Communication Networks (DRCN 2009), Washington, D.C., October 25-28, 2009.
- [42] G. Shen, W.D. Grover, "Extending the  $p$ -Cycle Concept to Path-Segment Protection," Proc. IEEE International Conference on Communications (ICC 2003), Anchorage, May 11-15, 2003, Session ON3, pp.1314-1319.
- [43] G. Shen, 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, vol. 21, no. 8, Oct. 2003, pp. 1306-1319.
- [44] I. N. Bronshtein, K. A. Semendyayev, Handbook of Mathematics, 3rd Edition, English Language translation by K. A. Hirsch, New York, NY: Springer-Verlag, 1985.
- [45] V. Chvatal, Linear Programming. Freeman and Co, 1983.
- [46] W. L. Winston, Operations Research Applications and Algorithms, 3rd Edition, Duxbury Press, Belmont, CA, 1994.
- [47] G. Birkan, J. Kennington, E. Olinick, A. Ortynski, G. Spiride. 2002. Making a Case for Using Integer Programming to Design DWDM Networks. Optical Networks Magazine. 4, 6 (2003) 107-120.
- [48] H. Kellerer; U. Pferschy, D. Pisinger, Knapsack Problems. Springer Verlag 2005.
- [49] G. Dantzig, Linear Programming and Extensions, Princeton, New Jersey, Princeton University Press, 1963.
- [50] A. H. Land, A. Doig, "An Automatic Method for Solving Discrete

- Programming Problems," *Electrometrica* 28, p497-520, 1960.
- [51] R. Gomory, "An Algorithm for the Mixed Integer Problem", Technical Report RM-2597, The Rand Corporation, 1960.
- [52] A. Caprara and M. Fischetti. Branch and cut algorithms. In M. Dell'Amico, F. Maoli, and S. Martello, ed., *Annotated Bibliographies in Combinatorial Optimization*, chapter 4. John Wiley, 1997.
- [53] ILOG, "ILOG CPLEX," accessed Feb 2009, [www.ilog.com/products/cplex](http://www.ilog.com/products/cplex), 2009.
- [54] R. Fourer, D. M. Gay, B. W. Kernighan, *AMPL: A Modeling Language For Mathematical Programming*, 2nd edition, Pacific Grove, California, Brooks/Cole-Thompson Learning, 2003.
- [55] E.K.P. Chong, S. H. Zak, *An Introduction to Optimization*, John Wiley & Sons, 2001.
- [56] E. W. Dijkstra, "A note on two problems in connection with graphs," *Number Math.*, vol. 1, 1959, pp. 269-271.
- [57] R. Bhandari, "Survivable Networks: Algorithms for Diverse Routing," Kluwer Academic Publishers, 1999.
- [58] D. B. Johnson, "Finding All the Elementary Circuits of a Directed Graph," *SIAM J. on Computing*, Vol. 4, 1975, pp. 77-84.
- [59] D. Morley, *Analysis and Design of Ring Based Transport Networks*, PhD Dissertation, University of Alberta, Spring 2001.
- [60] R. Freeman, *Fiber-Optic Systems for Telecommunications*. (New York: Wiley, 2002), Ch. 6 and 10.
- [61] D. Baloukov, W. D. Grover, and A. Kodian, "Toward Jointly Optimized Design of Failure-Independent Path-Protecting  $p$ -Cycle Networks," *J. Opt. Networking*. 7, 62-79 (2008).
- [62] B. Jaumard, C. Rocha, D. Baloukov, W. D. Grover "A Column Generation Approach for Design of Networks using Path-Protecting  $p$ -Cycles," *Proceedings of the 6th International Workshop on Design of Reliable Communication Networks (DRCN 2007)*, La Rochelle, France, 7-10 October 2007.
- [63] C. Ge, N. Bai, X. Sun, M. Zhang, "Iterative joint design approach for failure-independent path-protecting  $p$ -cycle networks," *J. Opt. Networking*. Vol. 6, No. 12, 1329-1339, Dec. 2007.
- [64] A. Grue, W. D. Grover, M. Clouqueur, D. Schupke, J. Doucette, B. Forst, D. Onguetou, D. Baloukov, "Comparative Study of Fully Pre-Cross-Connected Protection Architectures for Transparent Optical Networks," *Proceedings of the 6th International Workshop on Design of Reliable Communication Networks (DRCN 2007)*, La Rochelle, France, 7-10 October 2007.
- [65] A. Grue, B. Forst, D. Onguetou, D. Baloukov, J. Doucette, A.

- Kodian, W. D. Grover, "Comparative Study of Fully Pre-cross-connected Protection Architectures for High Availability Transparent Optical Networking: Summary Report on Progress November 2005-October 2006", Trlabs Tech. Rep. ST-06-01, Nov. 9, 2006.
- [66] A. Grue, B. Forst, D. Onguetou, D. Baloukov, W. D. Grover, "Comparative Study of Fully Pre-cross-connected Protection Architectures for High Availability Transparent Optical Networking: Project Year 2 Report", Trlabs Tech. Rep. ST-07-01, Nov. 30, 2007.
- [67] A. Grue, W. D. Grover, M. Clouqueur, D. Schupke, D. Baloukov, D. Onguetou, B. Forst, "CAPEX Costs of Lightly Loaded Restorable Networks Under a Consistent WDM Layer Cost Model," to appear in the proceedings of IEEE International Conference on Communications (ICC 2009), Dresden, Germany, June 14-18, 2009.
- [68] C. Barnhart, E. Johnson, G. Nemhauser, M. Savelsbergh, and P. Vance, "Branch-and-price: Column generation for solving huge integer programs," *Operations Research*, vol. 46, no. 3, pp. 316–329, May-June 1998.
- [69] W. D. Grover, J. Doucette, "Advances in optical network design with  $p$ -cycles: Joint optimization and pre-selection of candidate  $p$ -cycles," *Proc. IEEE LEOS Summer Topicals 2002*, Mont Tremblant, Québec, Canada, 15-17 Jul. 2002, pp. 49-50.
- [70] J. Doucette, "Advances on design and analysis of mesh-restorable networks," Ph.D. dissertation, University of Alberta, Edmonton, AB, Canada, 2004.
- [71] J. Doucette, D. He, W. D. Grover, and O. Yang, "Algorithmic approaches for efficient enumeration of candidate  $p$ -cycles and capacitated  $p$ -cycle network design," in *Proceedings of Fourth International Workshop on Design of Reliable Communication Networks (DRCN 2003)* (IEEE, 2003), pp. 212–220.
- [72] J. Doucette and W. Grover, "Influence of modularity and economy-of-scale effects on design of mesh-restorable DWDM networks," *IEEE J. Sel. Areas Commun.* 18, 1912–1923 (2000).
- [73] G. Shen, W. D. Grover, "Exploiting forcer structure to serve uncertain demands and minimize redundancy of  $p$ -cycle networks," *Proc. Fourth International Conference on Optical Networking and Communications (OptiComm 2003)*, Dallas, Texas, USA, 13-17 Oct. 2003.
- [74] J. Doucette, M. Clouqueur, W. D. Grover, "On the Availability and Capacity Requirements of Shared Backup Path-Protected Mesh Networks," *Optical Networks Magazine, Special Issue on Engineering the Next Generation Optical Internet*, vol. 4, no. 6, pp. 29-44, November/December 2003.
- [75] D. P. Onguetou, W. D. Grover, "Approaches to  $p$ -cycle network design with controlled optical path lengths on the restored network

- state," J. Opt. Networking. Vol. 7, No. 7, pp 673-691, July 2008
- [76] J. E. Doucette, P. A. Giese and W. D. Grover, "Combined Node and Span Protection Strategies with Node-Encircling  $p$ -Cycles," Proc. Workshop on Design of Reliable Communication Networks (DRCN), Ischia (Naples), Italy, pp. 213-221, 16-19 October 2005.
- [77] M. Gunkel, R. Leppla, M. Wade, A. Lord, D. Schupke, G. Lehmann, C. Fürst, S. Bodamer, B. Bollenz, H. Haunstein, H. Nakajima, J. Martensson, "A Cost Model for the WDM Layer", Photonics in Switching Conference (PS'2006), Oct., 2006.
- [78] IST project NOBEL, <http://www.ist-nobel.org>.
- [79] B. Wu, K. L. Yeung, P-H. Ho, "Formulations for  $p$ -Cycle Design without Candidate Cycle Enumeration", University of Hong Kong, Electrical and Electronic Engineering Department, Tech. Rep. TR-2008-001, [http://www.eee.hku.hk/research/doc/tr/TR2008001\\_IFDCC.pdf](http://www.eee.hku.hk/research/doc/tr/TR2008001_IFDCC.pdf)
- [80] A. Kodian, W.D. Grover, "Multiple-Quality of Protection Classes Including Dual-Failure Survivable Services in  $p$ -Cycle Networks," Proc. IEEE BROADNETS 2005, Boston, MA, USA.



# Chapter 10: Appendix

---

## 10.1 COST 239 [11]

### 10.1.1 Topology

NODE	X	Y	SIZE
N0	140	281	6
N1	315	350	4
N2	304	298	5
N3	356	235	5
N4	447	308	4
N5	417	161	5
N6	242	159	5
N7	250	214	5
N8	185	208	5
N9	128	143	4
N10	344	50	4

SPAN	O	D	LENGTH	UNITCOST
S1	N0	N1	820	820
S2	N0	N2	600	600
S3	N0	N5	1090	1090
S4	N0	N7	400	400
S5	N0	N8	300	300
S6	N0	N9	450	450
S7	N1	N2	320	320
S8	N1	N4	820	820
S9	N1	N8	930	930
S10	N2	N3	565	565
S11	N2	N4	730	730
S12	N2	N7	350	350
S13	N3	N10	740	740
S14	N3	N4	320	320
S15	N3	N5	340	340
S16	N3	N7	730	730
S17	N4	N5	660	660
S18	N5	N10	390	390
S19	N5	N6	660	660
S20	N6	N10	760	760
S21	N6	N7	390	390
S22	N6	N8	210	210
S23	N6	N9	550	550
S24	N7	N8	220	220
S25	N8	N9	390	390
S26	N9	N10	1310	1310

### 10.1.2 Demand Pattern 1

O	D	NBUNITS						
N0	N1	5	N0	N2	6	N0	N3	1
N0	N4	2	N0	N5	11	N0	N6	5
N0	N7	1	N0	N8	7	N0	N9	10
N0	N10	1	N1	N2	6	N1	N3	1
N1	N4	3	N1	N5	9	N1	N6	2
N1	N7	1	N1	N8	2	N1	N9	3
N1	N10	1	N2	N3	1	N2	N4	3
N2	N5	11	N2	N6	3	N2	N7	1
N2	N8	6	N2	N9	3	N2	N10	1
N3	N4	1	N3	N5	2	N3	N6	1
N3	N7	1	N3	N8	1	N3	N9	1
N3	N10	1	N4	N5	9	N4	N6	1
N4	N7	1	N4	N8	1	N4	N9	2
N4	N10	1	N5	N6	8	N5	N7	2
N5	N8	6	N5	N9	8	N5	N10	3
N6	N7	1	N6	N8	4	N6	N9	5
N6	N10	1	N7	N8	1	N7	N9	1
N7	N10	1	N8	N9	4	N8	N10	1
N9	N10	1						

### 10.1.3 Demand Pattern 2

O	D	NBUNITS						
N0	N1	4	N0	N2	4	N0	N7	4
N1	N2	4	N1	N7	4	N1	N8	4
N2	N5	4	N2	N9	4	N3	N7	4
N3	N8	4	N4	N7	4	N5	N6	4
N6	N7	4	N6	N8	4	N7	N10	4

## 10.2 Germany Network [10]

### 10.2.1 Topology

NODE	X	Y	SIZE
N01	380	521	2
N02	247	521	3
N03	100	516	3
N04	219	393	6
N05	17	461	3
N06	414	413	2
N07	340	456	4
N08	408	271	2

N09	301	262	3
N10	121	334	4
N11	213	263	5
N12	408	213	2
N13	383	120	2
N14	291	66	3
N15	307	158	2
N16	241	164	2
N17	116	76	4

SPAN	O	D	LENGTH	UNITCOST
S01	N01	N02	120	120
S02	N01	N07	232	232
S03	N02	N03	95	95
S04	N02	N04	100	100
S05	N03	N04	133	133
S06	N03	N05	255	255
S07	N04	N05	246	246
S08	N04	N07	183	183
S09	N04	N10	215	215
S10	N04	N11	262	262
S11	N05	N10	145	145
S12	N06	N07	31	31
S13	N06	N08	30	30
S14	N07	N09	73	73
S15	N08	N09	34	34
S16	N09	N11	152	152
S17	N10	N11	294	294
S18	N10	N17	229	229
S19	N11	N12	71	71
S20	N11	N17	187	187
S21	N12	N13	53	53
S22	N13	N14	62	62
S23	N14	N15	72	72
S24	N14	N17	156	156
S25	N15	N16	119	119
S26	N16	N17	149	149

### 10.2.2 Demand Pattern

O	D	NBUNITS						
N01	N03	2	N01	N11	2	N02	N03	1
N02	N04	1	N03	N04	3	N03	N05	1
N03	N07	1	N03	N08	1	N03	N09	1
N03	N10	2	N03	N11	1	N03	N14	1
N03	N16	1	N04	N05	2	N04	N07	4
N04	N08	1	N04	N09	4	N04	N10	3
N04	N11	5	N04	N14	1	N04	N17	2

N05	N10	2	N05	N11	1	N05	N16	1
N05	N17	1	N06	N07	1	N06	N08	1
N06	N09	1	N07	N08	1	N07	N09	4
N07	N10	3	N07	N11	3	N08	N09	2
N08	N11	1	N08	N14	1	N09	N10	3
N09	N11	5	N09	N14	1	N10	N11	3
N10	N14	1	N10	N15	1	N10	N16	1
N10	N17	1	N11	N12	1	N11	N13	1
N11	N14	2	N11	N15	2	N11	N16	2
N11	N17	1	N12	N13	1	N12	N14	1
N13	N14	1	N13	N16	1	N14	N15	1
N14	N16	1	N14	N17	1	N15	N16	1
N16	N17	1						

## 10.3 15n30s1 Network Family [70]

The data presented here for the 15n30s network family is based on the data found in [70].

### 10.3.1 Topology

NODE	X	Y	SIZE		
N01	125.000	140.000	3.000		
N02	268.000	55.000	4.000		
N03	413.000	162.000	4.000		
N04	525.000	97.000	3.000		
N05	290.000	233.000	6.000		
N06	612.000	218.000	5.000		
N07	508.000	349.000	4.000		
N08	572.000	485.000	4.000		
N09	402.000	456.000	4.000		
N10	259.000	325.000	6.000		
N11	291.000	598.000	3.000		
N12	292.000	483.000	4.000		
N13	186.000	458.000	3.000		
N14	49.000	421.000	4.000		
N15	75.000	274.000	3.000		
SPAN	O	D	LENGTH	UNITCOST	
S01	N01	N02	166.355	166.355	
S02	N01	N05	189.404	189.404	
S03	N01	N15	143.024	143.024	
S04	N02	N03	180.205	180.205	
S05	N02	N04	260.409	260.409	
S06	N02	N05	179.354	179.354	
S07	N03	N04	129.495	129.495	

S08	N03	N05	142.021	142.021
S09	N03	N06	206.729	206.729
S10	N04	N06	149.030	149.030
S11	N05	N07	246.941	246.941
S12	N05	N10	97.082	97.082
S13	N05	N15	218.874	218.874
S14	N06	N07	167.263	167.263
S15	N07	N08	150.306	150.306
S16	N07	N09	150.615	150.615
S17	N08	N06	269.980	269.980
S18	N08	N11	302.870	302.870
S19	N09	N08	172.456	172.456
S20	N09	N10	193.933	193.933
S21	N10	N06	368.860	368.860
S22	N10	N14	230.903	230.903
S23	N11	N12	115.004	115.004
S24	N11	N14	299.822	299.822
S25	N12	N09	113.265	113.265
S26	N12	N10	161.409	161.409
S27	N13	N10	151.717	151.717
S28	N13	N12	108.908	108.908
S29	N14	N13	141.908	141.908
S30	N15	N14	149.282	149.282

## 10.3.2 Family Members' Span Lists

15n30s1:	S01 S02 S03 S04 S05 S06 S07 S08 S09 S10 S11 S12 S13 S14 S15 S16 S17 S18 S19 S20 S21 S22 S23 S24 S25 S26 S27 S28 S29 S30
15n30s1-29s:	S01 S02 S03 S04 S05 S06 S07 S08 S09 S10 S11 S12 S13 S14 S15 S16 S17 S18 S19 S20 S21 S23 S24 S25 S26 S27 S28 S29 S30
15n30s1-28s:	S01 S02 S03 S04 S05 S06 S07 S08 S09 S10 S11 S13 S14 S15 S16 S17 S18 S19 S20 S21 S23 S24 S25 S26 S27 S28 S29 S30
15n30s1-27s:	S01 S02 S03 S04 S05 S06 S07 S09 S10 S11 S13 S14 S15 S16 S17 S18 S19 S20 S21 S23 S24 S25 S26 S27 S28 S29 S30
15n30s1-26s:	S01 S02 S03 S04 S05 S06 S07 S09 S10 S11 S13 S15 S16 S17 S18 S19 S20 S21 S23 S24 S25 S26 S27 S28 S29 S30
15n30s1-25s:	S01 S02 S03 S04 S05 S07 S09 S10 S11 S13 S15 S16 S17 S18 S19 S20 S21 S23 S24 S25 S26 S27 S28 S29 S30
15n30s1-24s:	S01 S02 S03 S04 S05 S07 S09 S10 S11 S13 S16 S17 S18 S19 S20 S21 S23 S24 S25 S26 S27 S28 S29 S30
15n30s1-23s:	S01 S02 S03 S04 S07 S09 S10 S11 S13 S16 S17 S18 S19 S20 S21 S23 S24 S25 S26 S27 S28 S29 S30
15n30s1-22s:	S01 S02 S03 S04 S07 S10 S11 S13 S16 S17 S18 S19 S20 S21 S23 S24 S25 S26 S27 S28 S29 S30
15n30s1-21s:	S01 S02 S03 S04 S07 S10 S11 S13 S16 S17 S18 S19 S20 S21 S23 S25 S26 S27 S28 S29 S30
15n30s1-20s:	S01 S02 S03 S04 S07 S10 S11 S13 S16 S17 S18 S20 S21 S23 S25 S26 S27 S28 S29 S30
15n30s1-19s:	S01 S02 S03 S04 S07 S10 S11 S13 S16 S17 S18 S20 S21 S23 S25 S26 S27 S29 S30
15n30s1-18s:	S01 S02 S03 S04 S07 S10 S11 S13 S16 S17 S18 S20 S21 S23 S25 S27 S29 S30
15n30s1-17s:	S01 S03 S04 S07 S10 S11 S13 S16 S17 S18 S20 S21 S23 S25 S27 S29 S30
15n30s1-16s:	S01 S03 S04 S07 S10 S11 S13 S16 S17 S18 S20 S23 S25 S27 S29 S30

## 10.3.3 Demand Pattern

O	D	NBUNITS						
N01	N02	8	N01	N03	4	N01	N04	5
N01	N05	6	N01	N06	10	N01	N07	3
N01	N08	9	N01	N09	9	N01	N10	5
N01	N11	10	N01	N12	1	N01	N13	1
N01	N14	4	N01	N15	4	N02	N03	8
N02	N04	2	N02	N05	5	N02	N06	1
N02	N07	6	N02	N08	1	N02	N09	1
N02	N10	3	N02	N11	9	N02	N12	6
N02	N13	5	N02	N14	5	N02	N15	7
N03	N04	2	N03	N05	9	N03	N06	2
N03	N07	3	N03	N08	6	N03	N09	5
N03	N10	9	N03	N11	9	N03	N12	6
N03	N13	1	N03	N14	4	N03	N15	2
N04	N05	3	N04	N06	2	N04	N07	4

N04	N08	7	N04	N09	9	N04	N10	3
N04	N11	5	N04	N12	6	N04	N13	4
N04	N14	2	N04	N15	1	N05	N06	10
N05	N07	7	N05	N08	9	N05	N09	6
N05	N10	3	N05	N11	3	N05	N12	1
N05	N13	1	N05	N14	7	N05	N15	3
N06	N07	4	N06	N08	6	N06	N09	4
N06	N10	5	N06	N11	2	N06	N12	5
N06	N13	8	N06	N14	9	N06	N15	3
N07	N08	5	N07	N09	5	N07	N10	7
N07	N11	7	N07	N12	1	N07	N13	9
N07	N14	4	N07	N15	7	N08	N09	2
N08	N10	3	N08	N11	6	N08	N12	8
N08	N13	8	N08	N14	2	N08	N15	10
N09	N10	8	N09	N11	6	N09	N12	8
N09	N13	10	N09	N14	1	N09	N15	1
N10	N11	10	N10	N12	2	N10	N13	1
N10	N14	7	N10	N15	1	N11	N12	5
N11	N13	7	N11	N14	1	N11	N15	3
N12	N13	3	N12	N14	5	N12	N15	1
N13	N14	10	N13	N15	3	N14	N15	3

