

**Design of Efficient Node  $p$ -Cycles  
in WDM Mesh Networks**

B. Jaumard, H. Li,  
C. Rocha

G-2012-31

June 2012



# Design of Efficient Node $p$ -Cycles in WDM Mesh Networks

**Brigitte Jaumard**

*GERAD & CSE  
Concordia University  
Montréal (Québec) Canada, H3G 1M8  
bjaumard@cse.concordia.ca*

**Honghui Li**

**Caroline Rocha**

*ECT  
Universidade Federal do Rio Grande do Norte (UFRN)  
Brazil*

June 2012

*Les Cahiers du GERAD*

G-2012-31

Copyright © 2012 GERAD



### Abstract

$p$ -Cycles have been extensively studied under a single link failure scenario. Even though not as common, single node failures may occur as well, and the resulting consequences can be devastating. Failure-Independent Path-Protecting (FIPP)  $p$ -cycles provide end-to-end path protection, and thereby protection against node failures, assuming that precautions are taken.

In this paper, we propose enhanced  $p$ -cycles, called node  $p$ -cycles, different from the FIPP  $p$ -cycle scheme, which can guarantee 100% protection against a single node failure. We design a scalable optimization model in order to design node  $p$ -cycles, which relies on a column generation formulation.

Extensive comparative experiments have been conducted after designing a column generation framework for all compared protection schemes, in order to guarantee scalable models for each of them. Experiments include (link)  $p$ -cycles, node  $p$ -cycles, FIPP  $p$ -cycles and a node  $p$ -cycle model previously proposed by Onguetou and Grover (2009). We show that node  $p$ -cycles offering node and link protection only require slightly more spare capacity than link  $p$ -cycles, while requiring sometimes less, sometimes more spare capacity than FIPP  $p$ -cycles. For the comparison with the work of Grover and Onguetou (2009), results show that our new node  $p$ -cycle scheme clearly outperforms their design in terms of capacity efficiency.

**Key Words:** Survivable WDM networks,  $p$ -cycles, node protection, column generation.

**Acknowledgments:** B. Jaumard was supported by NSERC (Natural Sciences and Engineering Research Council of Canada) and by a Concordia University Research Chair (Tier I) on the Optimization of Communication Networks.



# 1 Introduction

WDM techniques enable a single fiber to carry multiple non-overlapping channels, and each channel can operate at the speed of up to 100 Gbps. This translates to huge amounts of traffic carried on a WDM network. As optical WDM networks are prone to a single failure of network infrastructure (e.g., a fiber cut or a node failure), survivability is a critical requirement for WDM networks.

Various types of protection approaches have been proposed for ensuring the survivability of WDM mesh networks. Among these approaches,  $p$ -cycles (Pre-configured Pre-cross-Connected Protection Cycles) [1] hold the unique characteristic of ring-like recovery speed and mesh-like capacity efficiency. The last decade has seen numerous research studies on  $p$ -cycles. The overwhelming majority of these studies explored the protection of WDM networks against single link failures, the dominant failure scenario in WDM mesh networks, and did not worry about single node failures.

However, the failure of a single node can occur, even though not so often, as a result of disasters such as fires or floods. A single node failure is equivalent to the failure of all its incident fiber links and their connections through that node. The consequences of a node failure are therefore devastating.

In this paper, we investigate the design of  $p$ -cycle-based schemes for 100% node protection and propose a new scheme for node protection using  $p$ -cycles. The underlying idea comes from the observation that a node is protected if its two adjacent links on the working path are protected by the same  $p$ -cycle. It resembles the two hop approach of Grover and Onguetou [2], with one additional feature, detailed in Section 3. We develop a new Column Generation (CG) model to design  $p$ -cycles with node protection in order to adequately address the scalability issue afflicting the classical design of  $p$ -cycles. In order to make a quantitative comparison of the proposed node-protecting  $p$ -cycles with Failure-Independent Path-Protecting (FIPP)  $p$ -cycles [3], we adapt the CG model for the design of FIPP  $p$ -cycles of [4] so as to guarantee 100% protection against a single node or link failure.

Extensive experiments have been carried out over six network instances with diverse topology characteristics for the evaluation of the solution performances of four  $p$ -cycle-based schemes, including the newly proposed node-protecting  $p$ -cycles. Comparisons are made with respect to several parameters, such as required spare bandwidth, dual failure protection capability, lengths of the cycles.

The paper is organized as follows. In Section 2, we first explain how  $p$ -cycles can be used to guarantee node protection and then review the literature related to node protection in the context of  $p$ -cycle-based protection schemes. We conclude Section 2 with a comprehensive example for illustrating the distinctive characteristics of the four compared  $p$ -cycle-based protection schemes. In Section 3, we develop an original and scalable optimization model for the design of node  $p$ -cycles, i.e.,  $p$ -cycles ensuring 100 % node protection. In Section 4, we re-use the same optimization model framework to design a scalable model for the node  $p$ -cycle scheme of Grover and Onguetou [2]. In Section 5, we then adapt the column generation model of [4, 5] for the design of FIPP  $p$ -cycles in order to guarantee 100% node protection. In Section 6, we present the solution methods to the column generation models developed in Sections 3 to 5. Intensive comparative numerical results are presented in Section 7. Conclusions are drawn in Section 8.

## 2 Node $p$ -Cycles and Literature Review

### 2.1 Link $p$ -Cycles

We briefly recall below the original  $p$ -cycle protection scheme, which we will call link  $p$ -cycle in this paper in order to avoid confusion, due to the comparison of several  $p$ -cycle-based protection schemes.

Link  $p$ -cycles correspond to a shared link protection scheme with pre-configured pre-cross-connected structures. Upon the failure of a link, end nodes of the failing link switch the disrupted traffic automatically onto the  $p$ -cycle which protects the link, without any additional signaling requirement. Note that the protection of a given link, depending on the number of units to be protected, can be provided by more than one  $p$ -cycle. Thereby,  $p$ -cycles can achieve ring-like fast recovery speed.

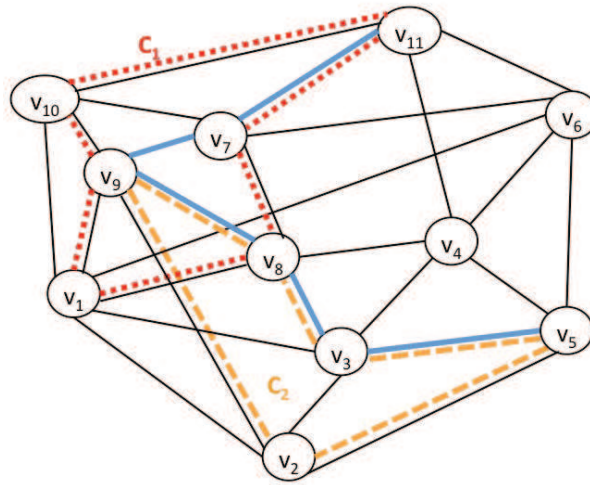


Figure 1: Overlapping  $p$ -cycles and node protection

An illustration of this scheme is provided in Figure 1. Therein, a working path ( $v_5 - v_3 - v_8 - v_9 - v_7 - v_{11}$ ) and two  $p$ -cycles, denoted by  $c_1$  ( $v_1 - v_9 - v_{10} - v_{11} - v_7 - v_8 - v_1$ ) and  $c_2$  ( $v_2 - v_9 - v_8 - v_3 - v_5 - v_2$ ), are drawn. As in ring-like networks, on-cycle links, such as  $\{v_7, v_{11}\}$  on  $p$ -cycle  $c_1$ , can be protected. Moreover, straddling links, such as link  $\{v_7, v_9\}$ , with two end nodes on  $p$ -cycle  $c_1$  but not lying on it, can also be protected. Thus,  $p$ -cycles can achieve mesh-like capacity efficiency.

## 2.2 Node $p$ -Cycles

In order to protect an intermediate node of a working path, a  $p$ -cycle can be constructed in such a way that the two adjacent links  $\ell$  and  $\ell'$  of a node  $v$  on a given working path  $p$  are protected by the same  $p$ -cycle. We denote such link pairs by  $(\ell, \ell')_{v,p}$ , or equivalently by  $\omega_p(v)$ , i.e., the co-cycle of  $v$  restricted to the links of  $p$ . Upon the failure of such an intermediate node, the two closest (adjacent) nodes on the working path can detect the failure and switch the disrupted traffic automatically onto the  $p$ -cycle(s), as in the link failure scenario.

An illustration is provided in Figure 1. In order to offer node and link protection to the request demand between nodes  $v_5$  and  $v_{11}$ , one needs two  $p$ -cycles  $c_1$  (short dashes) and  $c_2$  (long dashes). All links of the working path are protected as each of them is either an on-cycle link or a straddling link of  $c_1$  or  $c_2$ . Each intermediate node is also protected as the two adjacent links on its working path are protected by the same  $p$ -cycle. For instance, node  $v_7$  has its two adjacent links  $\{v_7, v_9\}$  and  $\{v_7, v_{11}\}$  protected by  $c_1$ . It follows that, if node  $v_7$  fails, the subpath  $v_9 - v_7 - v_{11}$  is rerouted on, e.g.,  $v_9 - v_{10} - v_{11}$ .

Node protection with  $p$ -cycles was previously investigated by Grover and Onguetou [2]. Their study relies on a 2-hop strategy which allows a  $p$ -cycle to protect a node with respect to one working path. Our proposal is more general as it allows the protection of a node lying on several working paths by the same  $p$ -cycle if the working paths use link disjoint protection entities following the failure of that node. As we will see in the numerical results, this allows reducing the spare bandwidth requirement.

Figure 2 illustrates the idea with a network topology comprising eight nodes and three demands routed on primary paths  $w_1$ ,  $w_2$  and  $w_3$ . Figure 2(a) shows the solution of the design strategy proposed in [2]. Therein, three  $p$ -cycles,  $c_1$ ,  $c_2$  and  $c_3$ , are required to provide full node protection. The resulting spare capacity requirement amounts to 12 channel units. However, with our node  $p$ -cycle scheme, as shown in Figure 2(b), only one  $p$ -cycle,  $c_4$ , is required in order to provide full node protection. Upon the failure of node  $v_1$ , on-cycle protection paths  $v_2 - v_6 - v_3$ ,  $v_3 - v_7 - v_4$  and  $v_4 - v_8 - v_5$  can be used to recover the three disrupted demands routed on  $w_1$ ,  $w_2$  and  $w_3$ , respectively. The associated spare capacity cost is eight units of channels. Compared with the strategy of [2], our solution reduces by 33% the amount of required bandwidth.

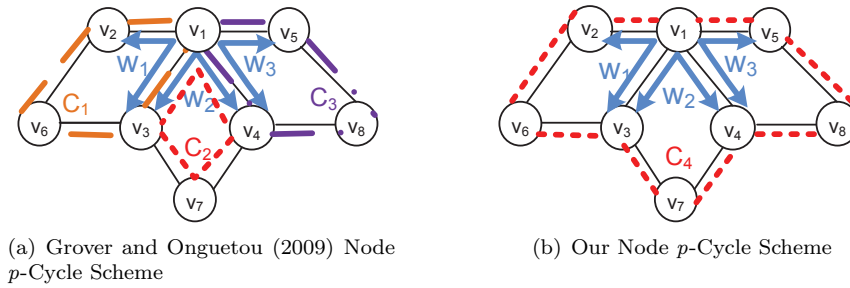


Figure 2: Comparison of our node  $p$ -cycle scheme with that of Grover and Onguetou (2009)

### 2.3 Literature Review

In order to provide node protection,  $p$ -cycles have been generalized to path-segment-protecting  $p$ -cycles (flow  $p$ -cycles for short) in [6], where a flow  $p$ -cycle can protect on-cycle and straddling working segments. Thus, the intermediate nodes of the working segments can be inherently protected but not the end nodes of the working segments. An improved version of segment  $p$ -cycles with the protection of the intermediate endpoints of the segments can be found in [7].

Kodian and Grover in [3] proposed Failure-Independent Path-Protecting (FIPP)  $p$ -cycles to provide end-to-end path protection. However, protection of all intermediate nodes is not necessarily guaranteed, see an example in Figure 3(a) (extracted from the example on Figure 3 in [3] with  $v_1 \equiv A, v_2 \equiv B, v_3 \equiv C, v_4 \equiv G, v_5 \equiv E$  and a shortened FIPP  $p$ -cycle). Therein, the  $p$ -cycle is represented by dashed lines, and the two requests with plain lines:  $v_1 - v_2 - v_3$  and  $v_2 - v_4$ . These requests cannot be both protected against the failure of  $v_2$  as their working paths are not node disjoint, while protected by the same FIPP  $p$ -cycle. However, if node disjointness is required instead of only link disjointness for all the paths to be protected by a single unit FIPP  $p$ -cycle, 100% node protection (excluding the endpoints of the working paths) can be guaranteed. Note that flow  $p$ -cycles and FIPP  $p$ -cycles are usually more capacity efficient than link protection schemes [6, 3]. However, optical recovery is slower as for any path or segment protection schemes.

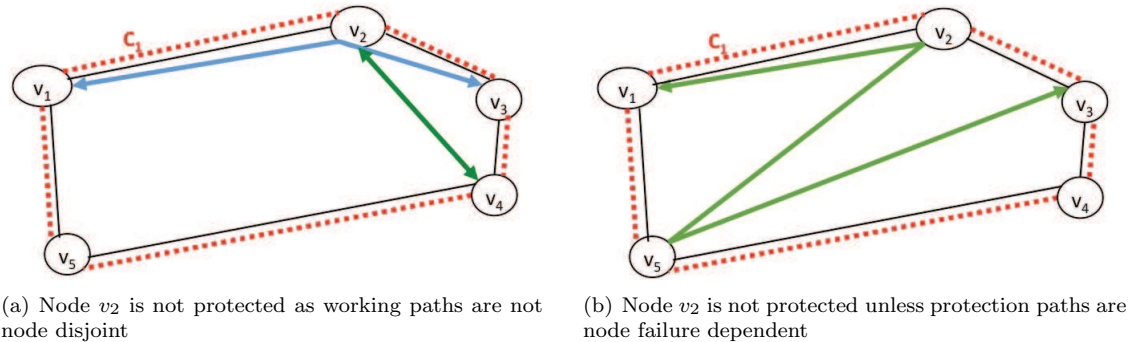


Figure 3: Two cases where node protection is not guaranteed by FIPP  $p$ -cycles

Given the above facts, one may wonder whether it is possible to provide full node protection using the original link  $p$ -cycles without necessarily going up to the FIPP  $p$ -cycles. A few studies have investigated this question. Stamatelakis and Grover [8] proposed node-encircling  $p$ -cycles (NEPCs), which are rather spare capacity consuming. Schupke [9] proposed and evaluated an Automatic Protection Switching (APS) protocol enhancement to provide means for node protection with  $p$ -cycles. Onguetou and Grover [10] proposed a new insight of node protection with overlapping  $p$ -cycles providing partial or full protection of all the intermediate nodes of the working paths. The same authors later restrict in [2] the segments to be two-hop segments in order to retain the simplicity of  $p$ -cycle switching operations.

In most previous works, comparisons were performed with heuristics as the classical model for the design of  $p$ -cycle-based schemes relies on an Integer Linear Program (ILP) model where each variable is associated with a potential cycle. Then, the solution of such an ILP model involves the off-line generation of either the whole set of potential  $p$ -cycles [1, 9, 11], or a restricted set of promising candidate  $p$ -cycles [12, 13], leading to a huge ILP or to a heuristic solution with unknown accuracy, respectively. Indeed, it has been shown, in the case of FIPP  $p$ -cycles [14], that heuristics may provide quite inaccurate solutions in some cases.

Most studies on FIPP  $p$ -cycles did not pay attention to node protection, but merely to path versus link protection with respect to bandwidth requirements, as in [3, 14, 5, 15]. FIPP  $p$ -cycle design may vary from one study to the next (see [4] for a thorough analysis of their differences). Indeed, as a path protection scheme, FIPP  $p$ -cycles can inherently provide node protection assuming protection and working paths are pairwise node disjoint, and the set of working paths which are protected by the same unit FIPP  $p$ -cycle are node disjoint. An illustration of that condition is shown in Figure 3(b). Observe that the working path  $v_1 - v_2 - v_5 - v_3$  can be protected against the node failure of  $v_2$  and  $v_5$  only if the protection path differs from one failure to the next, i.e., is node failure dependent.

To simplify notations and to improve the readability of the paper, we will talk about link, segment, path  $p$ -cycles instead of  $p$ -cycles, flow  $p$ -cycles and FIPP  $p$ -cycles. Link  $p$ -cycles with node protection capability will be referred to as node  $p$ -cycles.

## 2.4 Motivation and a Comprehensive Example

Path  $p$ -cycles are known to be more capacity efficient than link  $p$ -cycles in the context of protection against a single link failure [5, 15]. With respect to single node failures, to the best of our knowledge, comparisons can only be found in [2], where the conclusion is that node  $p$ -cycles hold a capacity efficiency comparable to path  $p$ -cycles and, therefore, are an attractive approach for link and node protection. Our conclusions are slightly different, depending on the data instances. The motivation of this paper is to go one step further with our enhanced node  $p$ -cycle scheme on the one hand, and exact solution of the ILP models on the other hand, rather than heuristic solutions or exact solutions on very small network and traffic instances.

Before going into the details of the optimization models, we propose to first explore the distinctive characteristics between the various protection schemes on a small example. Extensive computational experiments will be done in Section 7. We use the BELLCORE network topology with two unit demand requests, see Figure 4(a). Therein, the number on each link is the link geographical length [16], expressed in thousands of kilometers. Three protection schemes are illustrated: link  $p$ -cycles in Figure 4(b), i.e., the classical  $p$ -cycle protection scheme, node  $p$ -cycle in Figure 4(c) as per our newly proposed node  $p$ -cycle scheme described in Section 2.2, and path  $p$ -cycle in Figure 4(d), i.e., the so-called FIPP  $p$ -cycle with the required adaptations in order to guarantee node protection, as discussed in Section 5. Note that, on that simple example, both our node  $p$ -cycle protection scheme and that of Grover and Onguetou give the same results. For differences, see the example illustrated in Figure 2 or to Section 7.

The protection schemes, which are illustrated in Figure 4, can be described as follows. To protect traffic against a single link failure, two link  $p$ -cycles,  $c_1$  and  $c_2$  as shown in Figure 4(b) are used, with a spare capacity requirement of 14.5. To provide protection not only against single link failures, but also against single node failures, we need again two node  $p$ -cycles,  $c_3$  and  $c_4$ , different from  $c_1$  and  $c_2$ , as depicted in Figure 4(c). Their overall spare capacity is equal to 31.5. If, instead of link  $p$ -cycles, we now consider path  $p$ -cycles, again two  $p$ -cycles  $c_5$  and  $c_6$  are needed, as shown in Figure 4(d), with a spare capacity equal to 38.8.

This example shows that, as expected, the node  $p$ -cycles require more spare capacity than the link  $p$ -cycles for node protection. On the other hand, while it would have been expected that node  $p$ -cycles might have required more spare bandwidth than path  $p$ -cycles, it is not the case for the above example. As will be seen in the sequel, node  $p$ -cycles are sometimes more bandwidth efficient, sometimes less bandwidth efficient than path  $p$ -cycles, depending on the network topologies.

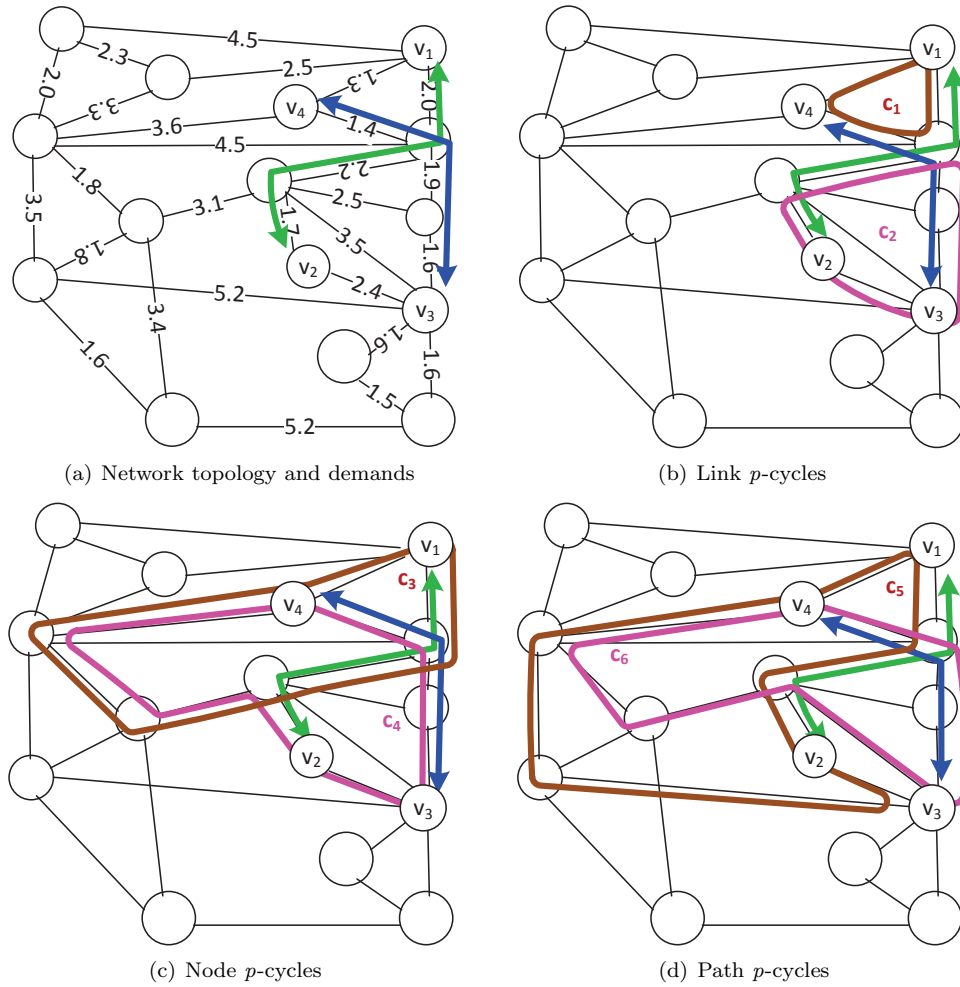


Figure 4: Illustration of different  $p$ -cycle-based protection schemes

### 3 A New Scalable Optimization Model for Node $p$ -Cycles

#### 3.1 Notations

Let us represent a WDM mesh network by a graph  $G = (V, L)$ , where  $V$  is the set of nodes indexed by  $v$ , and  $L$  is the set of fiber links indexed by  $\ell$ . Let  $d_\ell$  be the number of traffic units routed on link  $\ell$ . For a given working path  $p \in P$ , let  $D_p$  be the number of connection requests carried on it, and let  $V_p^*$  be the set of its intermediate nodes.

We propose a model for the design of  $p$ -cycle-based schemes against any single link or node failures with minimum bandwidth requirements. The model relies on  $p$ -cycle configurations, where a  $p$ -cycle configuration consists of a one unit  $p$ -cycle and the set of links and nodes it protects.

Before setting the details of the model, we need to introduce some notations.

**Sets**

- $C$  set of  $p$ -cycle configurations, indexed by  $c$ .
- $\omega(V')$  co-cycle of  $V' \subseteq V$ , i.e., the set of links with exactly one end node belonging to  $V'$ . Note that a co-cycle corresponds to a cut in an undirected graph.
- $\omega(v)$  set of links adjacent to node  $v$  (particular case of  $\omega(V')$  when  $V' = \{v\}$ ).

**Parameters**

$\text{COST}_\ell$	cost of link $\ell$
$\text{COST}_c$	cost of configuration $c$ , defined by $\text{COST}_c = \sum_{\ell \in c} \text{COST}_\ell$ .
$a_\ell^c \in \{0, 1, 2\}$	number of working capacity units on link $\ell$ protected by $p$ -cycle $c$ . $a_\ell^c = 1$ (resp. 2) if link $\ell$ is an on-cycle (resp. a straddling) link, 0 otherwise.
$a_{pv}^c \in \{0, 1\}$	number of protection paths provided by $p$ -cycle $c$ for working path $p$ against the failure of node $v$ .

**Variables**

$z^c$	number of copies of configuration $c$ .
-------	---

We can observe that by assuming  $a_{pv}^c \in \{0, 1, 2\}$  rather than  $\{0, 1\}$ , we can easily embed some additional cases for node protection as illustrated in Figure 5. Indeed, if a node  $v$  belonging to a working path is on a 2-hop chord of the  $p$ -cycle, then the latter one can provide 2 units of protection to that node. Such a node will be called a *straddling* node. Note that the links adjacent a straddling node will be protected together or one at a time by another  $p$ -cycle(s) as, in this case, they cannot be protected by the same  $p$ -cycle than node  $v$ . For instance, in Figure 5, demand  $v_1 - v_5$  is routed on path  $v_1 - v_2 - v_3 - v_4 - v_5$ , with  $p$ -cycle  $c_1$  built to protect node  $v_3$ . If  $v_3$  fails, its adjacent nodes  $v_2$  and  $v_4$  can switch the disrupted traffic on  $c_1$ , which can offer two protection paths. However, note that neither link  $v_2 - v_3$  nor  $v_3 - v_4$  can be protected by  $c_1$ . We included straddling nodes in our formulations in order to facilitate the comparisons with the node protection scheme of Grover and Onguetou [2]. However, we observed that very few such nodes are encountered in the solutions, most probably due to the fact that the protection of their adjacent links requires another  $p$ -cycle (possibly two more).

From now on, we will therefore assume that  $a_{pv}^c \in \{0, 1, 2\}$ .

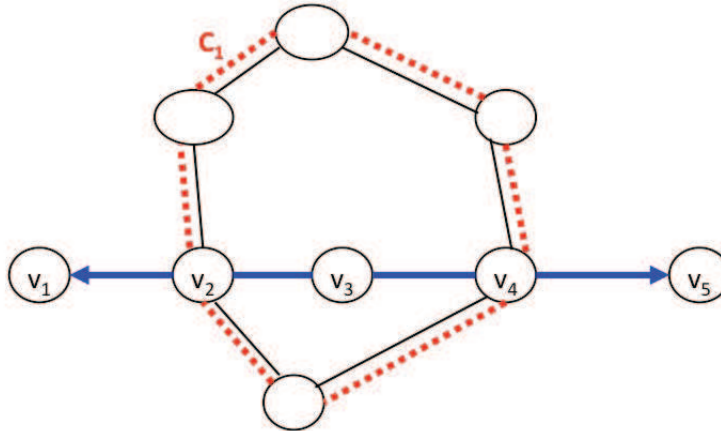


Figure 5: A straddling node

### 3.2 An Enhanced Optimization Model for Node $p$ -Cycles

We propose an optimization method based on Column Generation (CG) techniques for the design of node  $p$ -cycles according to the description given in Section 2. Following a CG modelling, the design problem is decomposed into two subproblems: the master problem and the pricing problem. The master problem selects the best combination of  $p$ -cycles in order to guarantee the node/link protection while the pricing problem dynamically generates new  $p$ -cycles, which iteratively contribute to the improvement of the current value of the objective of the (continuous relaxation of the) master problem. This way, the number of generated cycles remain reasonable without jeopardizing the solution optimality. See Section 6 for the details.

### 3.2.1 The master problem

The master problem relies on the concept of configurations, where a  $p$ -cycle configuration  $c$  is represented by a vector  $(a_\ell^c)_{\ell \in L}$  and a matrix  $(a_{pv}^c)_{p \in P, v \in V_p^*}$ .

The mathematical model can then be written as follows.

$$\min \sum_{c \in C} \text{COST}_c z^c \quad (1)$$

subject to:

$$\sum_{c \in C} a_\ell^c z^c \geq d_\ell \quad \ell \in L \quad (2)$$

$$\sum_{c \in C} a_{pv}^c z^c \geq D_p \quad p \in P, v \in V_p^* \quad (3)$$

$$z^c \in \mathbb{Z}^+ \quad c \in C. \quad (4)$$

Constraints (2) ensure that the overall traffic is protected against a single link failure. Constraints (3) ensure that all demands are protected against a single failure at node  $v$  with respect to working path  $p$ , for all intermediate nodes on all working paths. Constraints (4) define the domains of variables  $z^c$ .

### 3.2.2 The pricing problem

The role of the pricing problem is to generate a promising  $p$ -cycle that, once added to the master problem, will improve the value of the current solution of the linear relaxation of the master problem. Its objective corresponds to the minimization of the so-called reduced cost subject to a set of constraints for the generation of not only a  $p$ -cycle but also of the set of its protected links/nodes. Readers who are not familiar with linear programming are referred to, e.g., Chvatal [17].

The objective function can be written as follows.

$$\min \quad \text{COST}_c - \sum_{\ell \in L} u_\ell a_\ell^c - \sum_{p \in P} \sum_{v \in V_p^*} u_{pv} a_{pv}^c \quad (5)$$

where  $u_\ell$  and  $u_{pv}$  are the dual variables associated with constraints (2) and (3) respectively.

Before presenting the set of constraints of the pricing problem, we need to introduce the following notations.

#### Parameters

$v_{pv}^1, v_{pv}^2$  the two nodes adjacent to node  $v$  on path  $p$ .

#### Variables

$s_\ell = 1$  if link  $\ell$  straddles the cycle under construction; 0 otherwise.

$b_\ell = 1$  if  $\ell$  is an on-cycle link of the cycle under construction, 0 otherwise.

$y_v = 1$  if node  $v$  lies on the cycle under construction; 0 otherwise.

$x_{pv}^\ell = 1$  if link  $\ell$  is used to protect working path  $p$  against the failure of intermediate node  $v$ ; 0 otherwise.

The objective function of the pricing problem can then be re-written as follows:

$$\min \quad \underbrace{\sum_{\ell \in L} \text{COST}_\ell b_\ell}_{\text{COST}_c} - \sum_{\ell \in L} u_\ell \underbrace{(b_\ell + 2s_\ell)}_{a_\ell^c} - \sum_{p \in P} \sum_{v \in V_p^*} u_{pv} \underbrace{\sum_{\ell \in \omega(v_{pv}^1)} x_{pv}^\ell}_{a_{pv}^c}. \quad (6)$$

The set of constraints can be subdivided into two groups of constraints. The first group of constraints is associated with the generation of a simple cycle and the identification of the set of links which are protected by this cycle. The second group takes care of determining pairs  $(p, v)$ , made of a working path  $p$  and one of its intermediate node  $v$ , such that node  $v$  is protected by the cycle with respect to  $p$ .

The first group of constraints can be written as follows:

$$\sum_{\ell \in \omega(v)} b_\ell = 2 y_v \quad v \in V \quad (7)$$

$$s_\ell \leq y_v - b_\ell \quad v \in V, \ell \in \omega(v) \quad (8)$$

$$s_\ell \geq y_v + y_{v'} - b_\ell - 1 \quad v, v' \in V, \ell = \{v, v'\} \in L \quad (9)$$

$$\sum_{\ell \in \omega(V')} b_\ell \geq y_v + y_{v'} - 1 \quad 3 \leq |V'| \leq |V| - 3, V' \subset V, \quad (10)$$

$$v \in V', v' \in V \setminus V'.$$

Each node on a given cycle must have two incident links on that cycle. This is ensured by constraints (7). Constraints (8) and (9) are used to identify straddling links. A link straddles a cycle if its two end nodes are on the cycle, but the link itself is not. Constraints (10) prevent from generating a  $p$ -cycle configuration which includes multiple cycles. Otherwise, the identification of the straddling links would become too difficult.

Before describing the second group of constraints, we introduce one more notation:  $P_v^*$ , the set of working paths going through node  $v \in V$ , with  $v$  being an intermediate node of the path.

$$x_{pv}^\ell \leq b_\ell \quad p \in P, v \in V_p^*, \ell \in L \quad (11)$$

$$x_{pv}^\ell = 0 \quad p \in P, v \in V_p^*, \ell \in \omega(v) \quad (12)$$

$$\sum_{\ell \in \omega(v_{pv}^1)} x_{pv}^\ell = \sum_{\ell \in \omega(v_{pv}^2)} x_{pv}^\ell \quad p \in P, v \in V_p^* \quad (13)$$

$$\sum_{\ell \in \omega(v')} x_{pv}^\ell \leq 2 \quad p \in P, v \in V_p^*, v' \in V \quad (14)$$

$$\sum_{\ell \in \omega(v') | \ell \neq \ell'} x_{pv}^\ell \geq x_{pv}^{\ell'} \quad p \in P, v \in V_p^*, v' \in V \setminus \{v_{pv}^1, v_{pv}^2\}, \ell' \in \omega(v') \quad (15)$$

$$\sum_{p \in P_v^*} x_{pv}^\ell \leq 1 \quad \ell \in L, v \in V \quad (16)$$

$$b_\ell, s_\ell \in \{0, 1\} \quad \ell \in L \quad (17)$$

$$y_v \in \{0, 1\} \quad v \in V \quad (18)$$

$$x_{pv}^\ell \in \{0, 1\} \quad \ell \in L, p \in P, v \in V_p^*. \quad (19)$$

Constraints (11) ensure that only on-cycle links are eligible for protecting an intermediate or a straddling node of working path  $p$  against a single node failure. Constraints (12) say that, if a link is adjacent to an intermediate node  $v$  of working path  $p$ , the link cannot be used for protection of node  $v$ , i.e., cannot be on a protection path going around node  $v$ . Constraints (13) - (15) are flow conservation constraints for defining the protection paths. Constraints (13) say that a protection path, which protects node  $v$ , must go through the two nodes which are adjacent to node  $v$  on working path  $p$ . Constraints (13), together with (14), ensure that a  $p$ -cycle can provide at most two protection paths for the protection of a node lying on a working path. Constraints (14) and (15) ensure that protection paths are simple paths (without any loop), meaning that, a given node is either not on the protection path, or only encountered once. Upon the failure of a node, working paths passing through the failing node are all disrupted. Constraints (16) say that a link channel can only be used for recovering one unit of disrupted working paths. The last sets of constraints (17), (18), and (19) define the domains of the variables.

## 4 A Column Generation Reformulation of the Model of Grover and Onguetou (2009)

In order to compare our node  $p$ -cycle scheme with the one by Grover and Onguetou (2009) [2], we next rewrite their integer linear program in a column generation (CG) framework so as to obtain a scalable model which can solve larger instances than in their original paper. Note that instead of a two step process as in [2] (firstly, identify the best  $p$ -cycles, secondly, decide for each node which  $p$ -cycle protects it), the proposed CG model corresponds to an equivalent one step solution process.

We first recall their original ILP model with unified notations, and then reformulate it in a column generation framework.

### 4.1 ILP Model by Grover and Onguetou (2009)

While re-using notations of the previous section, we need two additional sets of variables, which are defined as follows:

- $z_{pv}^c$  number of copies of  $p$ -cycle  $c$  to protect intermediate node  $v$  of path  $p$ .
- $t_{pv}^c$  number of demand units of node  $v$  in path  $p$ , which are protected by  $p$ -cycle  $c$  in the event of a failure of node  $v$ .

The mathematical model of [10] can be written as follows (with unified notations):

$$\min \sum_{c \in C} \text{COST}_c z^c \quad (20)$$

subject to:

$$\sum_{c \in C} a_\ell^c z^c \geq d_\ell \quad \ell \in L \quad (21)$$

$$t_{pv}^c \leq a_{pv}^c z_{pv}^c \quad p \in P, v \in V_p^*, c \in C \quad (22)$$

$$\sum_{c \in C} t_{pv}^c \geq D_p \quad p \in P, v \in V_p^* \quad (23)$$

$$z_{pv}^c \leq M a_{pv}^c \quad p \in P, v \in V_p^*, c \in C \quad (24)$$

$$\sum_{p \in P_v} z_{pv}^c \leq z^c \quad v \in V, c \in C \quad (25)$$

$$z^c \in \mathbb{Z}^+ \quad c \in C \quad (26)$$

$$z_{pv}^c, t_{pv}^c \in \mathbb{Z}^+ \quad c \in C, p \in P, v \in V_p^*. \quad (27)$$

The objective is to minimize the spare capacity requirements. Constraints (21) ensure 100% guaranteed protection against a single link failure. Constraints (22) determine the units of protection that a  $p$ -cycle can provide for a single intermediate or straddling node of a working path upon the node failure. Constraints (23) ensure that all intermediate nodes of all working paths are protected against a single node failure. Constraints (24) indicate that, a  $p$ -cycle can be used to protect an intermediate (resp. straddling) node if  $(\ell, \ell')_{v,p}$  link pair is on the cycle (resp. straddle the cycle, as illustrated in Figure 5). Constraints (25) say that, for a given path  $p$  going through node  $v$ , if node  $v$  is protected by  $p$ -cycle  $c$  with respect to  $p$ , the reserved copies of unit  $p$ -cycle  $c$  should be enough to recover all impaired traffic on  $p$  against node  $v$  failure. Constraints (26) and (27) define the domains of the variables.

Set of constraints (21)–(27) can be equivalently rewritten in the following more compact way, reducing the number of variables (eliminating variables  $t_{pv}^c$ ), and eliminating the constraints (24) with the  $M$  constraint, which usually slow down the solution of the linear programming relaxation.

$$\sum_{c \in C} a_\ell^c z^c \geq d_\ell \quad \ell \in L \quad (28)$$

$$\sum_{c \in C} a_{pv}^c z_{pv}^c \geq D_p \quad p \in P, v \in V_p^* \quad (29)$$

$$\sum_{p \in P_v} a_{pv}^c z_{pv}^c \leq z^c \quad v \in V, c \in C \quad (30)$$

$$z^c \in \mathbb{Z}^+ \quad c \in C \quad (31)$$

$$z_{pv}^c \in \mathbb{Z}^+ \quad c \in C, p \in P, v \in V_p^*. \quad (32)$$

## 4.2 CG Reformulation of the Model by Grover and Onguetou (2009)

In order to reformulate the above ILP in a column generation framework, the design problem is decomposed into two problems which will be solved alternately: the master problem and the pricing problem. The master problem selects  $p$ -cycles which are calculated on the fly by the pricing problem at each iteration of the CG algorithm. The  $p$ -cycle candidates are associated with configuration set  $C$ . A configuration  $c$  consists of a cycle that protects a set of links and a set of intermediate (possibly straddling) nodes of working paths. Unlike the  $p$ -cycle configuration defined in Section 3.2.1, a node  $p$ -cycle configuration can only be used to protect a given node for a single impaired working path.

The CG reformulation leads to a master problem, which is the same as the model (1)–(4), i.e., our newly proposed model. However, while the objective of the pricing problem is the same as (5), its constraints differ. Before presenting it, we need to introduce the following sets and variables.

### Set

- $\omega_p(v)$  Pair of links on a working path  $p$  which are adjacent to an intermediate or a straddling node  $v$ . Those link pairs will be denoted by  $(\ell, \ell')_{v,p}$  in the sequel.
- $V_{p,v}$  Pair of nodes that are adjacent to node  $v$  on working path  $p$ .

### Variables

- $b_p^v =$  1 if the cycle under construction protects node  $v$  in path  $p$ , with  $v$  lying on the cycle; 0 otherwise.
- $s_p^v =$  1 if the cycle under construction protects node  $v$  in path  $p$ , with  $v$  being a straddling node; 0 otherwise.

Using these notations, the objective (5) can then be rewritten as follows.

$$\min \sum_{\ell \in L} \overbrace{\text{COST}_\ell}^{\text{COST}_c} b_\ell - \sum_{\ell \in L} u_\ell \overbrace{(b_\ell + 2s_\ell)}^{a_\ell} - \sum_{p \in P} \sum_{v \in V_p^*} u_p^v \overbrace{(b_p^v + 2s_p^v)}^{a_{pv}^c}. \quad (33)$$

Again, the pricing problem contains two blocks of constraints. The first one consists of the constraints for defining a cycle, which are identical to constraints (7) to (10). The second block of constraints is as follows:

$$b_p^v \leq b_\ell + s_\ell \quad p \in P, v \in V_p^*, \ell \in \omega_p(v) \quad (34)$$

$$s_p^v \leq y_{v'} \quad p \in P, v \in V_p^*, v' \in V_{p,v} \quad (35)$$

$$s_p^v \leq 1 - y_v \quad p \in P, v \in V_p^* \quad (36)$$

$$\sum_{p \in P_v} (b_p^v + s_p^v) \leq 1 \quad v \in V \quad (37)$$

$$b_\ell, s_\ell \in \{0, 1\} \quad \ell \in L \quad (38)$$

$$y_v \in \{0, 1\} \quad v \in V \quad (39)$$

$$b_p^v, s_p^v \in \{0, 1\} \quad p \in P, v \in V_p^*. \quad (40)$$

Constraints (34) say that an intermediate node on a working path  $p$  can be protected in an on-cycle manner, i.e., the node belongs to the cycle under construction, if its two adjacent links on  $p$  are also protected by the cycle under construction. Constraints (35) and (36) together say that the cycle under construction can protect a straddling node  $v$  on a working path  $p$  if its two adjacent nodes on  $p$  belongs to the cycle (i.e., are on-cycle nodes), but not  $v$  itself. Upon node failure of  $v$ , all working paths transmitting the failed node are disrupted, but the cycle under construction can offer node protection only for a single path with respect to node  $v$ , which is stated by constraints (37).

This last CG-ILP model, through its pricing problem, is not fully equivalent to the model of Grover and Onguetou [10], but corresponds to where they aim at with their two phase process. Details follow.

Firstly in the above pricing problem of the CG-ILP model, we directly generate a unique  $p$ -cycle for the protection of a given node, as obtained with the two step process in [2]. Indeed, a candidate  $p$ -cycle consists of a unit cycle, the protected links and the *potential* protected intermediate nodes of working paths, in such a way that a node can be protected by a pcycle for a unique working path. As shown in Figure 6, two candidates,  $c_1$  and  $c_2$  are generated respectively for protecting node  $v$  of path  $p_1$  and for  $v$  of  $p_2$ . The master problem then determines how many units of  $c_1$  and  $c_2$  are required in the solution.

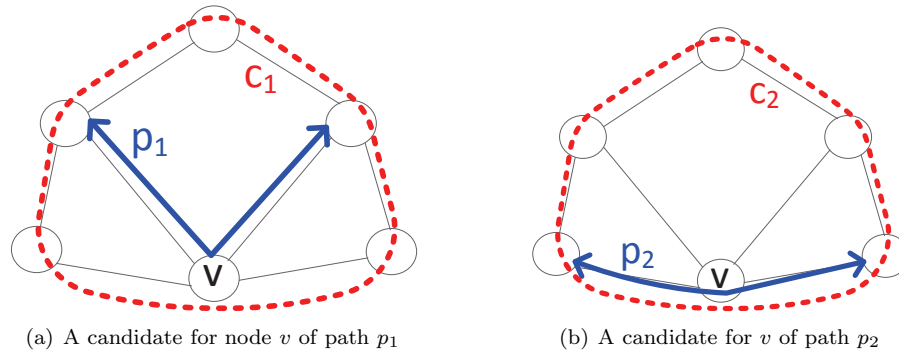


Figure 6: Example candidates for the CG model

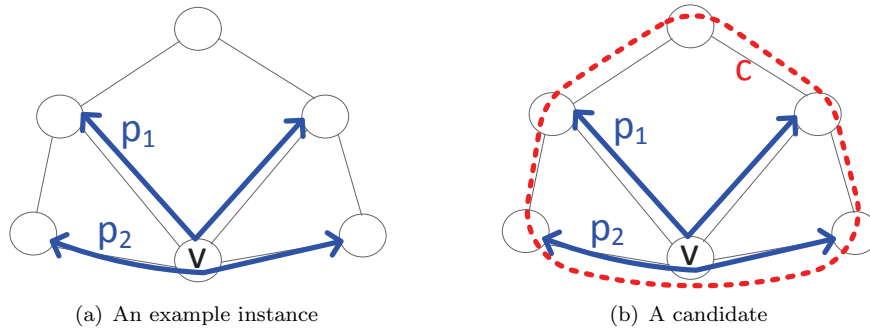


Figure 7: An example candidate for the ILP model by Grover and Onguetou

Figure 7(a) presents a network topology, and two demands carried respectively on paths  $p_1$  and  $p_2$ . Both paths pass through node  $v$ . Figure 7(b) illustrates a candidate  $p$ -cycle, which is made up of an unit cycle  $c$ , the protected (on-cycle and straddling) links and the potential protected node  $v$  of  $p_1$  and of  $p_2$ . The CG-ILP model then determines in turn how many units of cycle  $c$  are allocated for node  $v$  of  $p_1$  and for node  $v$  of  $p_2$ .

## 5 Column Generation Model for the Design of FIPP $p$ -Cycles

We re-use the column generation formulation of [4] and adapt it in order to guarantee 100% node protection. Note that we consider here, as in [4], a decomposition of the pricing problem in order to further enhance the

CG model scalability as path  $p$ -cycles are more costly (with respect to computing times) to compute than link  $p$ -cycles.

As for the models in Sections 3 and 4, the master problem selects path  $p$ -cycles for protection of all working paths against single link or node failures so that the overall spare capacity is minimized. A configuration  $c$  is made of a cycle and the set of working paths it protects against single node/link failures, represented by vector  $a^c$ . The vector element  $a_p^c \in \{2, 1, 0\}$  encodes the number of protection paths provided by this configuration  $c$  for working path  $p$ . The mathematical model is next formulated.

$$\min \sum_{c \in C} \text{COST}_c z^c$$

subject to:

$$\sum_{c \in C} a_p^c z^c \geq D_p \quad p \in P \quad (41)$$

$$z^c \in \mathbb{Z}^+ \quad c \in C. \quad (42)$$

Constraints (41) express that all working paths should be protected against a single link or node failure. Constraints (42) define the domains of the variables.

The goal of the pricing problem is to generate a valid augmenting configuration, i.e., a unit path  $p$ -cycle with the set of requests it protects against single node/link failures. We are discussing below the changes to the previous model of [4] so to guarantee protection against single node failures.

The first one is due to the fact that link disjoint paths are not necessarily node disjoint, as shown in Figure 8. Therein, node  $v_7$  can not be protected if both working paths  $v_3 - v_{11}$  (blue path) and  $v_1 - v_{10}$  (green path) are protected by the path  $p$ -cycle  $c_1$  (in red in Figure 8). The second one deals with the case of (partially) straddling working paths with one intermediate node belonging to a path  $p$ -cycle. For instance, see working path  $v_3 - v_{11}$  (blue path) in Figure 8. Indeed, for protection against single link failures, it is required that working and backup paths to be pairwise link disjoint. Similarly, in order to guarantee node protection, working and backup paths must be pairwise node disjoint. Back to the example with working path  $v_3 - v_{11}$ , path  $p$ -cycle  $c_1$  offers only one unit of protection through the backup path  $v_3 - v_4 - v_{11}$  as one of its intermediate node,  $v_9$ , is on the cycle, even if the working path is a straddling one.

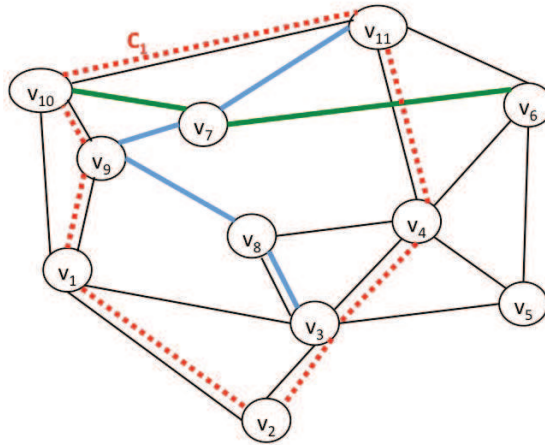


Figure 8: Link disjoint paths which are not node disjoint

The pricing problem generates a promising configuration for the master problem in each iteration of the CG algorithm such that the value of the current solution (linear relaxation) may be improved. Its objective corresponds to the minimization of the so-called reduced cost of the linear relaxation subject to the configuration constraints, and can be written as follows.

$$\overline{\text{COST}}_c = \text{COST}_c - \sum_{p \in P} u_p a_p^c \quad c \in C, \quad (43)$$

where  $u_p$  are the dual variables related to constraints (41).

Before describing the constraints of the pricing problems, we need to define some additional notations. Previously defined notations still hold.

### Parameters

$\beta^{p,p'}$  = 0 if working paths  $p$  and  $p'$  are node disjoint (and hence, link disjoint); 1 otherwise.

$\tau_\ell^{p,v}$  = 1 if link  $\ell$  is adjacent to an intermediate node  $v$  of working path  $p$ ; 0 otherwise.

$v_p^1, v_p^2$  = end nodes of working path  $p$ .

### Variables

$x_\ell^p$  = 1 if link  $\ell$  is used to protect working path  $p$ ; 0 otherwise.

Using these notations, the reduced cost (43) can then be rewritten:

$$\overline{\text{COST}}_c = \sum_{\ell \in L} \overbrace{\text{COST}_\ell}^{\text{COST}_c} b_\ell - \sum_{p \in P} u_p \sum_{\ell \in \omega(v_p^1)} \overbrace{x_\ell^p}^{a_p^c}$$

Once more, constraints can be subdivided into two subsets. The first subset takes care of the generation of a cycle:

$$\sum_{\ell \in \omega(v)} b_\ell \leq 2 \quad v \in V \quad (44)$$

$$\sum_{\ell' \in \omega(v): \ell' \neq \ell} b_{\ell'} \geq b_\ell \quad \ell \in \omega(v), v \in V \quad (45)$$

$$\sum_{\ell \in \omega(V')} b_\ell \geq b_{\ell'} + b_{\ell''} - 1 \quad V' \subseteq V, \ell' \in \omega(V'), \ell'' \in L, \ell'' \notin \omega(V'). \quad (46)$$

Constraints (44) and (45) ensure that, for an on-cycle node, there exist two on-cycle links adjacent to it. Constraints (46) prevent producing a configuration which includes multiple cycles, as otherwise identification of (partially) straddling paths is difficult.

The second subset of constraints deals with the identification of the protected working paths by the current cycle.

$$x_\ell^p \leq b_\ell \quad p \in P, \ell \in L \quad (47)$$

$$x_\ell^p \leq 1 - \tau_\ell^{p,v} \quad p \in P, v \in V_p^*, \ell \in \omega_p(v) \quad (48)$$

$$\sum_{\ell \in \omega(v_p^1)} x_\ell^p = \sum_{\ell \in \omega(v_p^2)} x_\ell^p \quad p \in P \quad (49)$$

$$\sum_{\ell \in \omega(v)} x_\ell^p \leq 2 \quad p \in P, v \in V \quad (50)$$

$$\sum_{\ell' \in \omega(v): \ell' \neq \ell} x_{\ell'}^p \geq x_\ell^p \quad p \in P, v \in V, \ell \in \omega(v) \quad (51)$$

$$x_\ell^p + x_\ell^{p'} \leq 2 - \beta^{p,p'} \quad p, p' \in P, \ell \in L \quad (52)$$

$$b_\ell \in \{0, 1\} \quad \ell \in L \quad (53)$$

$$x_\ell^p \in \{0, 1\} \quad p \in P, \ell \in L. \quad (54)$$

Note that Constraints (52) can be equivalently rewritten:

$$\sum_{p \in P_v^*} x_\ell^p \leq 1 \quad v \in V, \ell \in L,$$

with the advantage of reducing the number of constraints.

Constraints (47) say that only on-cycle links are eligible for protection of working paths. Constraints (48) ensure that a working path and its corresponding protection path are node disjoint. Constraints (49) say that, for any working path, its protection path(s) must end at its two end nodes. Constraints (49) and (50) ensure that the configuration under construction can provide at most two protection paths for any working path. Constraints (50) together with (51) guarantee flow conservation in the intermediate nodes along protection path(s). Constraints (52) ensure that only link-and-node-disjoint working paths can share protection paths.

## 6 Solution of the Optimization Models

We expose here the details of the solution process for all the optimization models described in the last three sections. The solution process includes two steps, as illustrated in Figure 9(a). The first step consists in solving the linear relaxation of the optimization model with the column generation (CG) technique. The inner loop (Step 1) illustrates the iterative process of the CG algorithm.

The outcome of the first step is a lower bound  $\underline{z} = z_{LP}^*$  (i.e., the optimal value of the linear relaxation) on the optimal integer solution of the model that is solved. The second step consists in building an integer solution of value  $\tilde{z}_{ILP}$ , as shown by the outer arrow (Step 2). In case an optimal solution is not reached ( $\tilde{z}_{ILP} > z_{ILP}^*$ ), the solution process provides the accuracy of the integer solution ( $\varepsilon = \frac{\tilde{z}_{ILP} - z_{ILP}^*}{z_{ILP}^*}$ ). As will be seen in Section 7, even if not all solutions are optimal, they are all  $\varepsilon$ -optimal with  $\varepsilon$  very small.

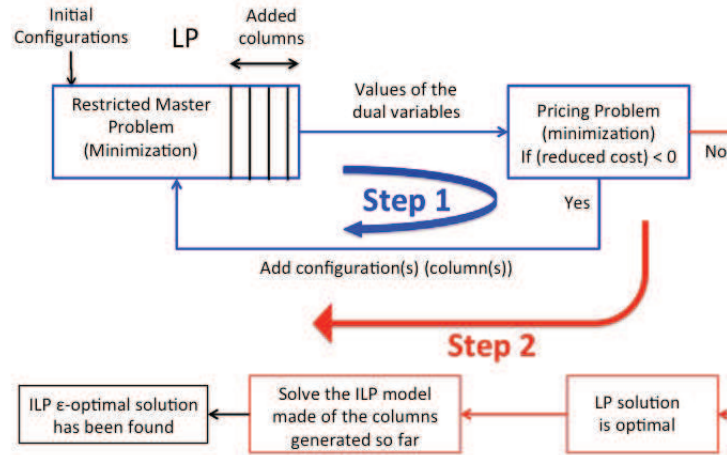
### 6.1 Column Generation Technique

We first briefly recall the classical column generation algorithm, and then explain the features we have added in order to enhance the original algorithm, while taking into account the characteristics of the models for the design of link or node or path  $p$ -cycles.

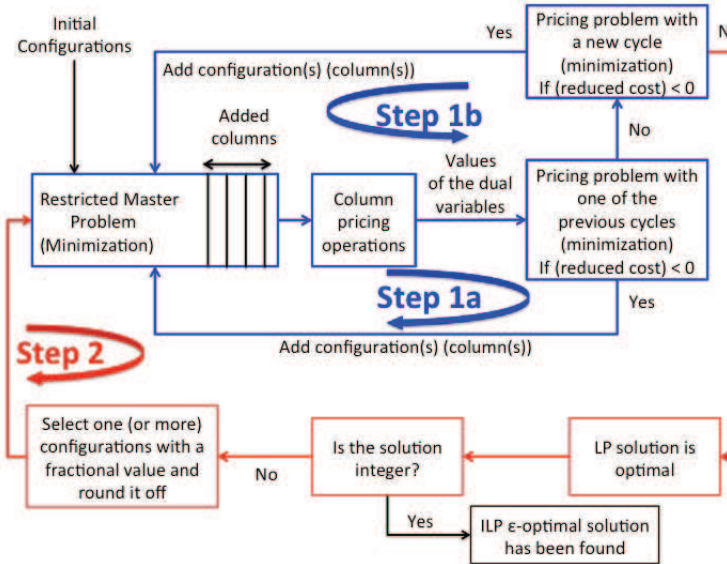
In order to deal with the optimization models arising in the design of  $p$ -cycle-based schemes, decomposition techniques are required in order to overcome their large number of variables. They allow an iterative solution scheme where (link/node/path)  $p$ -cycle configurations are on-line generated as long as there exists an augmenting configuration, i.e., a configuration such that its addition means an improvement of the current solution of the linear relaxation of the master problem.

In order to speed up the CG algorithm, based on the observation that we may have the same cycle belonging to different configurations, we first generate the augmenting configurations using the cycles which have already been used in the previously generated configurations. This iterative process is illustrated by Step 1a in Figure 9(b). Note that the search of a configuration is advantageously speeded up if a cycle has been already selected, meaning that the first group of constraints of the pricing problems disappear. If no augmenting configuration can be generated with the last generated cycles, then we solve the complete pricing problem, which includes the first group of constraints, i.e., the search of a cycle. This iterative process is illustrated by Step 1b in Figure 9(b). The resulting cycle can either be a brand new one, or one of the cycles generated with one of the earliest configurations. We observe, in practice, that very few complete pricing problems need to be solved, usually the number is of the order of the number of distinct cycles among the overall set of configurations.

Column generation algorithms which are described in the early papers often assume that we eliminate at each iteration the columns associated with nonbasic variables in order to keep the size of the constraint matrix as small as possible. However, in practice, this often entails regenerating several times some columns (or managing a column buffer), and therefore, the most efficient strategy is to “price out columns” only when the constraint matrix becomes too large, so say every 50 to 100 iterations in practice. Column pricing is



(a) Classical CG and ILP Solutions



(b) Enhanced CG and ILP Solutions

Figure 9: Solution of the optimization models

conducted looking at the variables/columns with the largest reduced cost. This technique corresponds to the box entitled “Column pricing operations” in Figure 9(b).

## 6.2 How to Get an Optimal (near Optimal) Integer Solution

Once the optimal solution of the linear relaxation of an ILP model has been reached, the next step consists in computing an integer solution, ideally an optimal one. However, in order to guarantee reaching an optimal integer solution, one must use a branch-and-price method (see, e.g., [18]) when a column generation model is used. It requires some effort in order to identify an efficient and scalable branching scheme. Instead, let  $\tilde{z}_{ILP}$  be the optimal integer solution of the ILP model such that its constraint matrix is associated with the last solved restricted master problem. It is well known that  $\tilde{z}_{ILP}$  is not necessarily the value  $z_{ILP}^*$  of an optimal integer solution of the master problem. However,  $z_{LP}^*$ , the optimal value of the linear relaxation of the master problem provides a lower bound on  $z_{ILP}^*$ . It follows that the accuracy  $\varepsilon$  of  $\tilde{z}_{ILP}$  can be measured

by the following optimality gap (in percentage):

$$\varepsilon = 100 \times \frac{\tilde{z}_{\text{ILP}} - z_{\text{LP}}^*}{z_{\text{LP}}^*}. \quad (55)$$

In this study, we used the so-called rounding up method. It consists in an iterative algorithm, which, at each iteration, selects the configuration variable with the largest fractional value, round it up, and set the variable to that rounded value. The linear relaxation of the resulting problem is updated, and then the algorithm iterates. This process is illustrated by Step 2 in Figure 9(b).

## 7 Computational Results

In this section, we evaluate the performances of the newly proposed scheme for node protection using  $p$ -cycles (node  $p$ -cycles), and compare it with three other schemes: (i) the OG  $p$ -cycle scheme of [2] (OG  $p$ -cycle) using a column generation reformulation as described in Section 4; (ii) the path  $p$ -cycle scheme using the model of Section 5; (iii) the link  $p$ -cycle scheme corresponding to the original  $p$ -cycles using the column generation model of [5]. Note that the column generation does not change the models initially proposed (in relation with the constraint set), but only the scalability of the solution process. In addition, they allow an exact and scalable solution of the models, and hence make their comparison more efficient and equitable.

All mathematical models have been implemented in C++ with their linear (restricted master problems) and ILP (pricing problems) programs solved by CPLEX 11.0.1 solver [19].

We estimate the bandwidth cost in two different ways, the number of link channels on the one hand, and the sum of the geographical distances of the links on the other hand, while assuming the cost to be proportional to these parameters. For performance evaluation and comparison, we use three metrics.

- Capacity redundancy -  $R^{p/w}$ , i.e., the ratio of spare capacity cost over working capacity cost.
- Dual link failure recovery ratio -  $R_2$  (as in Grover [20] (formula (8.3), p. 510) and Schupke [21]):

$$R_2 = 1 - \frac{\sum_{\ell, \ell' \in L \times L: \ell \neq \ell'} \text{LOSS}(\ell, \ell')}{\sum_{\ell, \ell' \in L \times L: \ell \neq \ell'} (d_\ell + d_{\ell'})}, \quad (56)$$

where  $d_\ell$  is the number of traffic units carried on link  $\ell$  (see notations in Section 3.1),  $\text{LOSS}(\ell, \ell')$  is the total number of unprotected traffic units under the dual failure of links  $\ell$  and  $\ell'$ .

- Number of topologically distinct  $p$ -cycles which need to be configured upon the deployment of a  $p$ -cycle scheme, a measure of the complexity of the protection management overhead.

### 7.1 Data Instances

We use six different network instances, which are described in Table 1, together with some key characteristics of the traffic instances. For each network instance, we provide the number of nodes, the number of edges, the nodal degree (average and standard deviation values) as an indicator of the regularity of the network connectivity, and the link length (minimum, average, maximum).

In order to go one step further in the analysis of the results, we added three more characteristics for the network instances. They are all expressed in two units, number of hops and km. We therefore computed the network diameter, i.e., the length of the longest path among the shortest paths for any node pair in the network. Moreover, for all working paths, we provide the average distance of the alternate shortest path, which is link and  $(\ell, \ell')_{v,p}$  link disjoint respectively, see the columns entitled Avg. alter. distance (resp. Avg. Link pair dist.).

At last, we added two parameters for characterizing the network connectivity in the context of path  $p$ -cycles and node  $p$ -cycles, which are defined below.

Table 1: Data instances

		Networks					
		NSF [22]	GERMANY [22]	BELLCORE [16]	NJLATA [16]	COST239 [23]	EON2004 [22]
#	nodes	14	17	15	11	11	28
	edges	21	26	28	23	26	41
Node degree	average	3.0	3.1	3.7	4.2	4.7	2.9
	deviation	0.6	1.2	1.3	2.1	0.6	0.9
Link length	min	387	36	130	250	210	218
	max	3,529	353	520	950	1,310	1,500
	avg.	1,303.5	170.3	262.5	553.9	578.7	625.4
Network diameter	[hops]	3	6	4	4	3	8
	[km]	5,316	951	1,160	1,800	1,660	5,051
Avg. alter. distance	[hops]	3.6	4.2	3.2	2.3	2.2	5.6
	[km]	4,181.9	722.7	886.6	1,179.8	996.3	3,277.9
Avg. link pair dist.	[hops]	3.7	3.5	2.9	2.2	2.2	3.8
	[km]	4,247.1	600.8	778.3	1,130.2	997.7	2,136.6
Network connectivity	DV	1.8	1.3	1.6	1.6	3.4	1.6
	DP	1.7	1.2	1.4	1.6	3.4	1.3
# Demands		91	136	105	55	55	378
Working cost		1,970	4,050	2,610	943	792	2,984

For a path  $p$ -cycle protection scheme, we define the  $DP$  parameter as the average number of node disjoint protection paths (for construction of path  $p$ -cycles) over the set  $\mathcal{SD}$  of node pairs, i.e., the set of node pairs with some traffic:

$$DP = \frac{\sum_{\{s,d\} \in \mathcal{SD}} DP_{sd}}{|\mathcal{SD}|}$$

where  $DP_{sd}$  is the maximum number of node disjoint protection paths for a node pair  $\{s, d\}$ .

For a node  $p$ -cycle protection scheme, we define the parameter  $DV$  as follows:

$$DV = \frac{\sum_{p \in P, v \in V_p^*} DV_{p,v} + \sum_{\ell \in L} DL_{\ell}}{\sum_{p \in P} |V_p^*| + |L|}$$

where  $DV_{p,v}$  and  $DL_{\ell}$  denote the maximum number of node disjoint protection paths for the  $(\ell, \ell')_{v,p}$  link pair and for link  $\ell$  (a one-hop working path), respectively.

Traffic instances are described by their number of demand requests and working capacity (i.e., number of link wavelength channels which are required for the primary paths), in the last two lines of Table 1. For each

network, the number of demand units between a given node pair is randomly generated (uniform distribution) in the interval  $[1..20]$  for the first five networks, and in the interval  $[1..3]$  for EON2004. Working paths are computed in such a way as to guarantee that they are of minimum length subject to the condition that there exists at least one potential protection path that is link and node disjoint with the working path.

## 7.2 Capacity Redundancy

We first compare the four protection schemes, link  $p$ -cycle (LpCycle), OG  $p$ -cycle (OGpCycle), node  $p$ -cycle (LNpCycle) and path  $p$ -cycle (PpCycle), with respect to capacity redundancy. Results are summarized in Figures 10(a) and 10(b) for the hop and the geographical distance (km) metrics, respectively. Except for Germany instance, the behavior (ranking of the four protection schemes according to the capacity redundancy) is identical for the two bandwidth cost metrics.

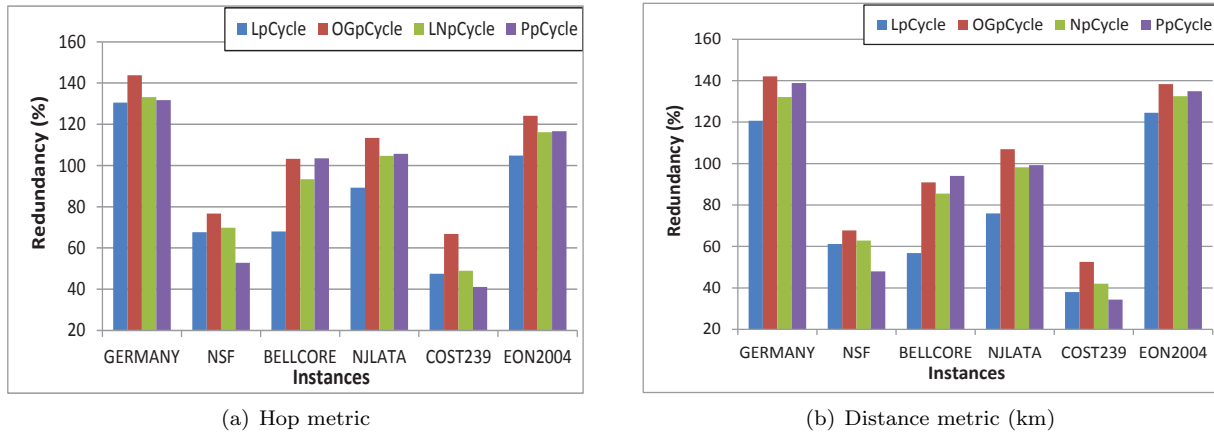


Figure 10: Capacity redundancies

As expected, extra spare capacity is required by  $p$ -cycles for node protection compared with those providing only link protection for each network instance. As for node  $p$ -cycle with hop metric (resp. for the geographical distance metric), from 1.3% to 25.3% (from 1.7% to 28.7%) more bandwidth is required than link  $p$ -cycle, while providing 100% protection against single link and node failures.

Node  $p$ -cycle is more capacity efficient than OG  $p$ -cycle for all network instances, keeping in line with the comparison of these two protection schemes in the example presented in Section 2 (see Figure 2). The redundancy differences vary from 7% to 18% for the hop metric, and from 5% to 11% for the geographical distance metric.

For path  $p$ -cycle vs. node  $p$ -cycle, for NSF and COST239 with the small deviation of node degree (0.6 shown in Table 1), the path  $p$ -cycle scheme has an advantage over the node  $p$ -cycle one in terms of capacity efficiency, following the belief that a shared path protection approach is more capacity efficient than a shared link one. The difference ranges from 7.8% to 17.1% for the channel metric, and from 7.7% to 14.9% for the geographical distance metric. However, for the four other instances with an high node degree deviation, node  $p$ -cycle outperforms path  $p$ -cycle, with respect to capacity efficiency. This is in accordance with the example illustrated in Section 2.4. The differences vary from 0.5% to 10.2% (resp. 1.1% to 8.6%) for the hop (resp. distance) metric. We analyze further those results in the next paragraph throughout the impact of the network topology.

### 7.2.1 Impact of the network topology

In order to understand the impact of the network topologies, we calculate the average distance of the shortest node disjoint protection paths for all working paths. The average distance is provided in the row with the header of “Avg. alter. distance” in Table 1. Also, for all  $(\ell, \ell')_{v,p}$  link pairs, we calculate the average distance of their shortest node-disjoint protection paths, see the row entitled by “Avg. link pair distance” in Table 1.

We can observe from Table 1 that, for NSF and COST239, the average path distance for the intermediate node protection is equal to or longer than that for the path protection. However, for the other four instances, the average path distance for the node protection is shorter than that for the path protection. Note that, in the optimal solution of node  $p$ -cycles (resp. path  $p$ -cycles), the shortest protection paths are most probably used for the construction of node  $p$ -cycles (resp. path  $p$ -cycles) in order to keep spare capacity usage minimal. As a result, node  $p$ -cycles require less spare capacity, and therefore are more capacity efficient than path  $p$ -cycles in these four instances. Thereby, the experimental results shown in Figure 10(b) are verified.

We next investigate further how the capacity efficiency of path  $p$ -cycles (resp. node  $p$ -cycles) varies with the topology characteristics, and on which kind of topology path  $p$ -cycles (resp. node  $p$ -cycles) can achieve better performance. To this end, we calculate the indicator DP (resp. DV) for each working path (resp. each  $(\ell, \ell')_{v,p}$  link pair with respect to an intermediate node of each working path), and present its value in Table 1. The relationship of DP (resp. DV) with the capacity redundancy is shown in Figure 11(a) (resp. 11(b)) for path  $p$ -cycles (resp. node  $p$ -cycles) respectively.

Figures 11(a) and 11(b) suggest that the average number DP (resp. DV) of alternate paths in a network is a key factor impacting the performance of path  $p$ -cycles (resp. node  $p$ -cycles) with respect to the capacity efficiency. In general, the more protection paths, the more efficient path  $p$ -cycles (resp. node  $p$ -cycles) are in the context of protection against a single link/node failure. The reason is as follows. For each working path (resp.  $(\ell, \ell')_{v,p}$  link pair), if more protection paths, and thereby more  $p$ -cycles are available, more opportunities then exist for the sharing of spare bandwidth with other node-disjoint working paths.

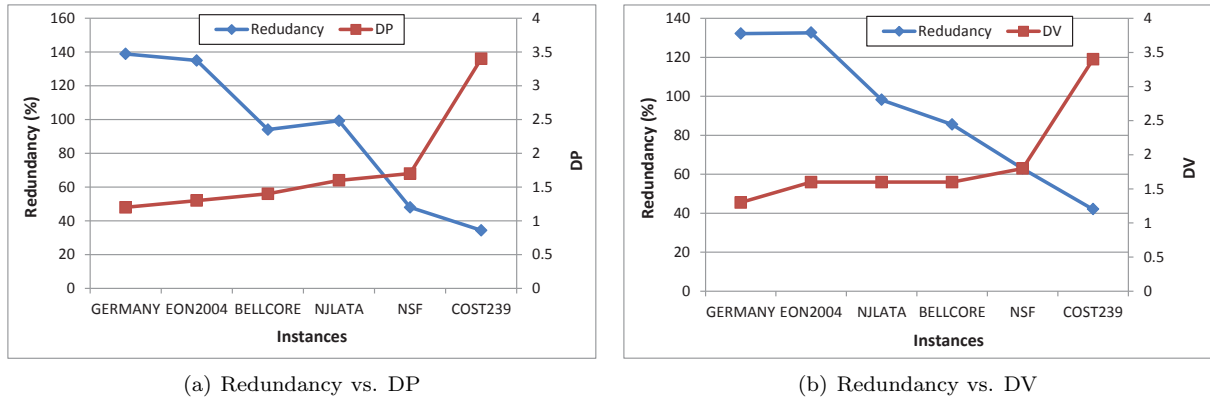


Figure 11: Capacity redundancies vs. connectivity

### 7.3 Number and Length of the Cycles

Assuming no restriction on the length of the  $p$ -cycles, in Figure 12, we present respectively the number and length of the cycles in the optimal solutions of link  $p$ -cycle (LpCycle), OG  $p$ -cycle (OGpCycle), node  $p$ -cycle (LNpCycle) and path  $p$ -cycle (PpCycle). Since the excessive length of  $p$ -cycles has been often criticized (see, e.g., [6, 24]), we will study, in Section 7.5, the impact of shortening the  $p$ -cycles on the performance metrics.

Here, we observe more differences when comparing the four protection schemes, depending on the selected bandwidth metric. With respect to the number of distinct cycles, comparative performance (ranking of the four protection schemes) is the same for BELLCORE, NJLATA and EON2004. With respect to the average length of the cycles (weighted with the number of cycle occurrences), comparative performance (the protection scheme ranking) is the same for the GERMANY, BELLCORE and EON2004 instance, while it differs for the other three instances. A first observation is therefore that the number of hops is not necessarily a good approximation of the geographical length depending on the network structure (the variance of the link length is not enough to predict the different ranking).

Overall, the link  $p$ -cycle scheme requires the smallest number of distinct cycles among the four protection schemes for all instances (independently of the topologies). For the distance metric, there is no clear winner

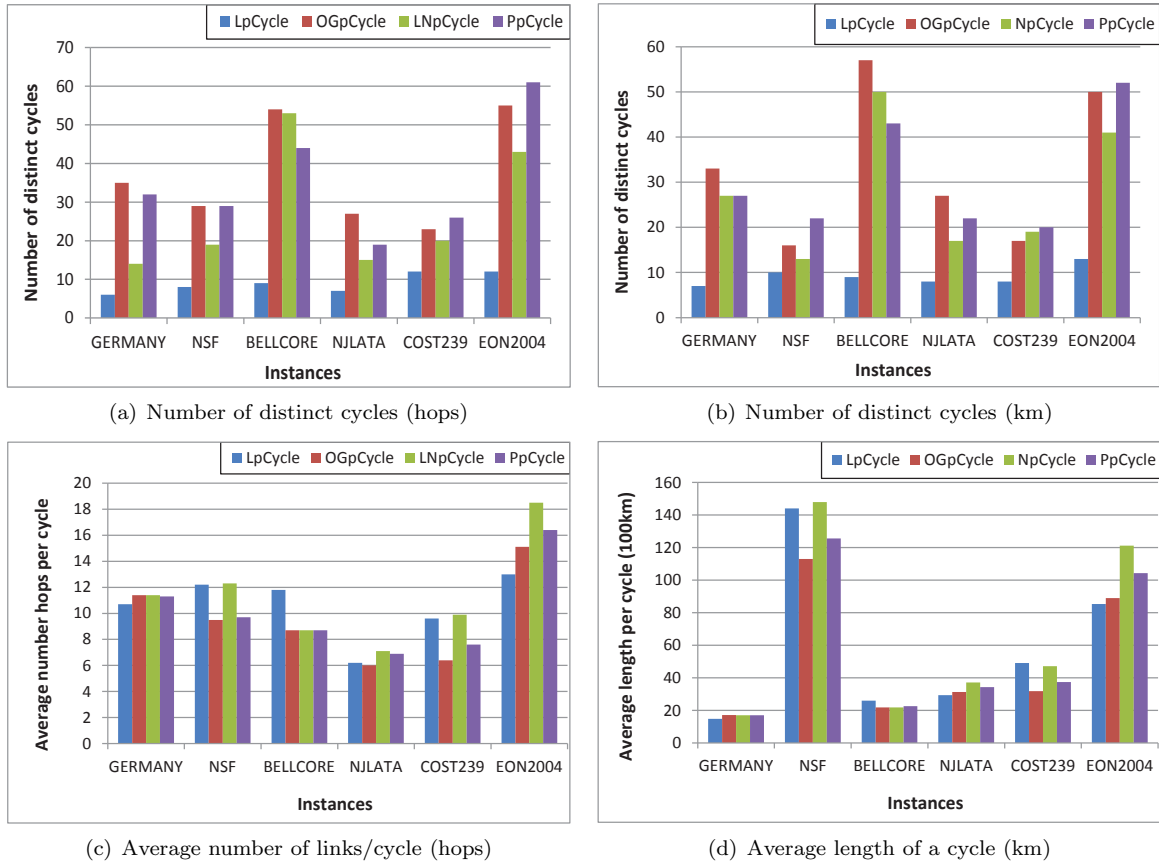


Figure 12: Number and Length of the Cycles

among the three other protection schemes for the number of distinct cycles, while for the hop metric, if we exclude the BELLCORE instance, the node  $p$ -cycle scheme requires the smallest number of distinct cycles, a possible advantage for the cycle management. Regarding OG  $p$ -cycle versus path  $p$ -cycle, the ranking varies with the network topologies, and there is no systematic ranking between them.

From the perspective of the average length of the  $p$ -cycles ensuring node protection, node  $p$ -cycle is equal to or larger than the other two protection schemes (OG  $p$ -cycle and path  $p$ -cycle). In contrast with link  $p$ -cycle, the average length of node  $p$ -cycle is larger in the GERMANY, NJLATA and EON2004 instances.

## 7.4 Protection against Dual Link Failures

We next compare the four protection schemes with respect to the dual link failure restoration ratio ( $R_2$ ) over the six network instances. Note that here the protection against dual link failures comes for free from the solutions ensuring 100% link or link/node protection. Results are displayed in Figures 13(a) and 13(b) with the hop and distance metrics respectively.

The ranking of the four schemes with the hop metric is similar to the one with the distance metric except for NSF and GERMANY instances. OG  $p$ -cycle holds the largest value of  $R_2$  for all instances except for GERMANY. This comes from the fact that OG  $p$ -cycle is the most capacity redundant among these four schemes. The  $R_2$  differences between node  $p$ -cycle and OG  $p$ -cycle range from 3.1% to 15% (resp. from 0.1% to 9.1%) for the hop metric (resp. the distance metric).

Except for EON2004 instance, node  $p$ -cycle outperforms path  $p$ -cycle. Node  $p$ -cycle can provide from 8% to 27.9% (resp. 6.5% to 22.4%) more  $R_2$  than path  $p$ -cycle for the hop (resp. distance) metric. In contrast with link  $p$ -cycle, the node  $p$ -cycle scheme provides comparable  $R_2$  value for all instances except for

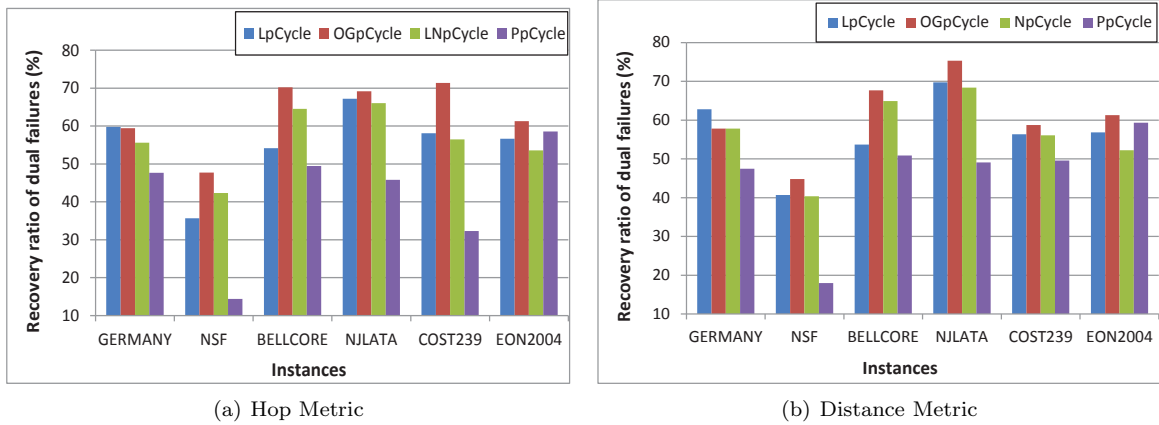


Figure 13: Traffic Weighted Dual Recovery Ratio

BELLCORE and NSF. This can be explained by the fact that the average length of the cycles in the solutions based on node  $p$ -cycle is larger than those based on link  $p$ -cycles. In BELLCORE, however, node  $p$ -cycles can provide 10% more  $R_2$  than link  $p$ -cycles for both bandwidth metrics.

## 7.5 Impacts of Cycle Length Limitation on Performance Metrics

The numerical results in the previous paragraphs have been obtained without any limitation on the  $p$ -cycle length (whether for link, node or path protection), as for most of the related studies in the literature. However, length restrictions may have to be taken into account, if the delay of a connection is limited, especially if we assume that wavelength conversion is available at each node (the assumption of this study). For those reasons, we next investigate further the performance of the node protection schemes (node  $p$ -cycle and path  $p$ -cycle), while some restrictions are applied on the length of the  $p$ -cycles. We added the following constraint in the pricing problems:

$$\sum_{\ell \in L} \text{LENGTH}_{\ell} b_{\ell} \leq \overline{\text{LENGTH}}, \quad (57)$$

where  $\overline{\text{LENGTH}}$  is the length limit on the  $p$ -cycles.

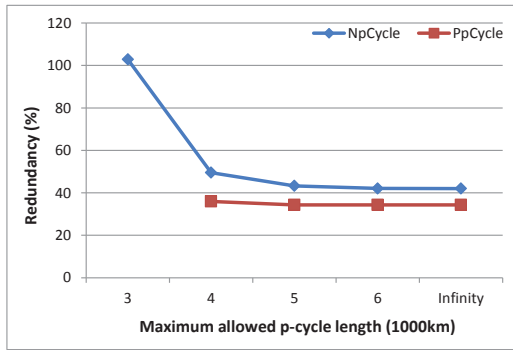
Experiments are carried out on the COST239 and EON2004 instances, and the numerical results are summarized in Figures 14 and 15 respectively.

For COST239 (resp. EON2004) network,  $\overline{\text{LENGTH}}$  value ranges between 3,000 km (resp. 7,000 km) and  $+\infty$  (no limit on the length of the  $p$ -cycles). Those values have been selected in order to guarantee that solutions exist for node  $p$ -cycle with 100% node protection. Note that for values slightly smaller, neither of these two instances can be fully protected by any of these two protection schemes.

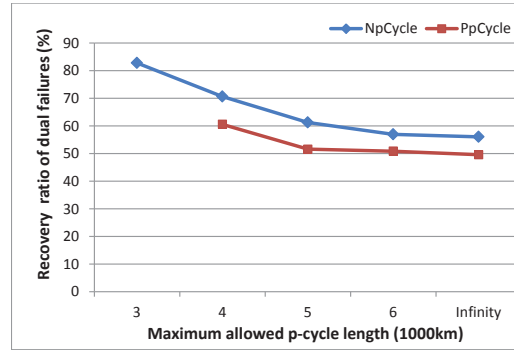
Figures 14(a) and 15(a) show the decrease of the redundancy ratio as the length of the  $p$ -cycles increases for COST239 and EON2004 respectively. Note that for both instances, especially, for EON2004, there is no path  $p$ -cycle protection ensuring full link/node protection much earlier than for the node  $p$ -cycle scheme ( $\overline{\text{LENGTH}} = 7,000$  vs. 12,000) as the length limit of  $p$ -cycles decreases. The reason is as follows.

For path  $p$ -cycles, the length limit on the cycle length should be at least twice as much as the network diameter (1,660km for COST239 and 5,051km for EON2004, as shown in Table 1) to ensure 100% guaranteed link and node protection for all demands. With respect to node  $p$ -cycle, however, there is no such restriction, and the length limit on cycles only needs to be larger than the size of any cycle ensuring 100% link protection. This implies that, node  $p$ -cycle is an interesting scheme for link and node protection in the context of the existence of very restricted length limit.

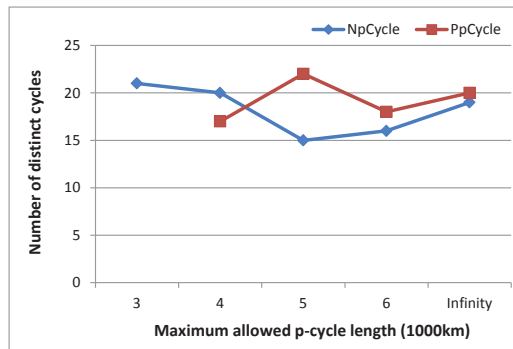
For both instances, there is a length threshold (5,000 for COST239 and 12,000 for EON2004) beyond which the redundancy of these two protection schemes does not decrease significantly.



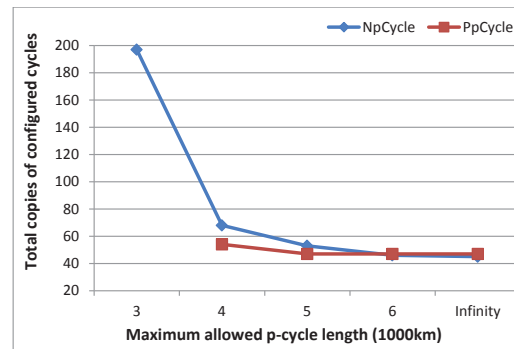
(a) Capacity redundancy



(b) Dual link failure restorability

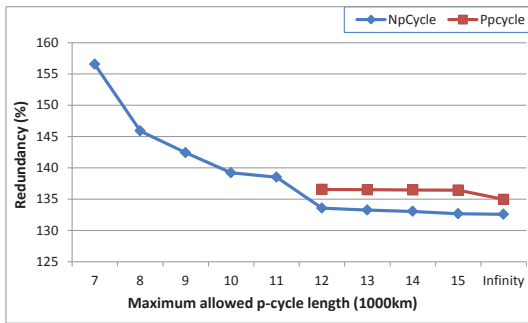


(c) Number of distinct cycles

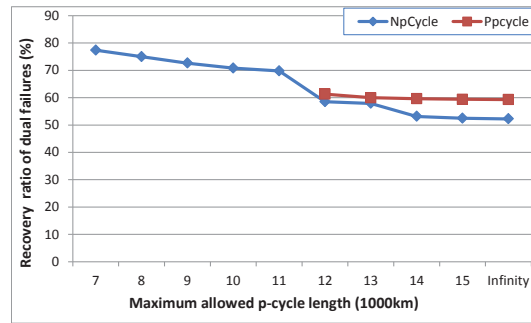


(d) Total copies of configured p-cycles

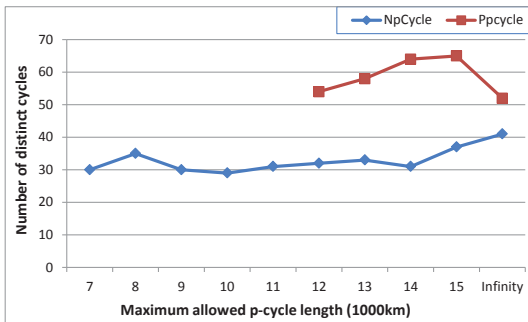
Figure 14: Effects of maximum allowed physical  $p$ -cycle length: COST239



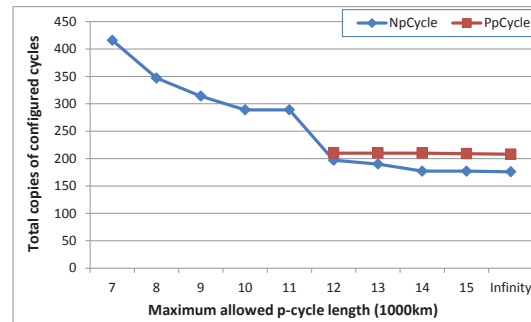
(a) Capacity redundancy



(b) Dual link failure restorability



(c) Number of distinct cycles



(d) Total copies of configured p-cycles

Figure 15: Effects of maximum allowed physical  $p$ -cycle length: EON2004

One of the advantages of short  $p$ -cycles, both for path and node  $p$ -cycles, but more so for node  $p$ -cycles, is the protection for dual link failures which reaches, e.g., 83% for node  $p$ -cycles when  $\overline{\text{LENGTH}} = 3,000$ , as shown in Figure 14(b). Although here, we evaluate the protection against dual failures which comes for free when ensuring protection against a single node/link failure, it shows that it should be rather easy to ensure protection against targeted dual failures (e.g., shared risk link group (SRLG) failures) while ensuring protection against a single link/node failure without requiring extra bandwidth.

For the overall number of  $p$ -cycles, there is a sharp increase of their number as the length limit decreases, especially for the node  $p$ -cycles, as shown in Figure 14(d) and 15(d). However, the number of distinct cycles (with respect to the links they use) remains rather stable. This suggests that the length limit has a minor impact on the management overhead.

## 8 Summary and Conclusions

In this paper, we have studied the comparison of four  $p$ -cycle-based schemes in WDM mesh networks for single link or node failures. It includes an enhanced node  $p$ -cycle scheme, which improves the first node  $p$ -cycle of Onguetou and Grover [2]. Unified column generation models have been written and developed for all compared protection schemes.

Numerical results show that ensuring node protection in addition to link protection against a single failure is not significantly more costly in terms of spare capacity than for link protection against a single failure.

Contrary to a wild belief that path  $p$ -cycles have an advantage of capacity efficiency over  $p$ -cycles for link and node protection, numerical results reveal that, in some network instances with high standard deviation for the node degree,  $p$ -cycles ensuring node protection hold comparable capacity efficiency as path  $p$ -cycles. In some network instances, node  $p$ -cycles are even more capacity efficient than path  $p$ -cycles. In addition, results show that  $p$ -cycles with node protection can achieve higher recovery ratio of dual link failures than path  $p$ -cycles. In the context of restricted length cycles, node  $p$ -cycle solutions exist for much smaller cycles than path  $p$ -cycle for full link and node protection. The above observations suggest that  $p$ -cycles with node protection are a promising alternative of path  $p$ -cycles for full link and node protection.

## References

- [1] W. D. Grover, D. Stamatelakis, Cycle-oriented distributed preconfiguration: Ring-like speed with mesh-like capacity for self-planning network restoration, in: IEEE International Conference on Communications - ICC, 1998, pp. 537–543.
- [2] W. Grover, D. Onguetou, A new approach to node-failure protection with span-protecting  $p$ -cycles, in: 11th International Conference on Transparent Optical Networks (ICTON), 2009, pp. 1–5.
- [3] A. Kodian, W. Grover, Failure-independent path-protecting  $p$ -cycles efficient and simple fully preconnected optical-path protection, *Journal of Lightwave Technology* 23 (10) (2005) 3241–3259.
- [4] C. Rocha, B. Jaumard, Efficient computation of FIPP  $p$ -cycles, to appear in *Telecommunications Systems*.
- [5] C. Rocha, B. Jaumard, Revisiting  $p$ -cycles / FIPP  $p$ -cycles vs. shared link / path protection, in: International Conference on Computer Communications and Networks - ICCCN, 2008, pp. 1–6.
- [6] G. Shen, W. Grover, Extending the  $p$ -cycle concept to path segment protection for span and node failure recovery, *IEEE Journal on Selected Areas in Communications* 21 (8) (2003) 1306–1319.
- [7] B. Jaumard, H. Li, Segment  $p$ -cycle design with full node protection in WDM mesh networks, in: 18th IEEE Workshop on Local Metropolitan Area Networks (LANMAN), 2011, pp. 1–6.
- [8] D. Stamatelakis, W. D. Grover, Theoretical underpinnings for the efficiency of restorable networks using pre-configured cycles, *IEEE Transactions on Communications* 48 (8) (2000) 1262–1265.
- [9] D. Schupke, Automatic protection switching for  $p$ -cycles in WDM networks, *Optical Switching and Networking* 2 (1) (2005) 35–48.
- [10] D. Onguetou, W. Grover, A new insight and approach to node failure protection with ordinary  $p$ -cycles, in: IEEE International Conference on Communications - ICC, 2008, pp. 5145–5149.
- [11] D. Schupke, C. Gruber, A. Autenrieth, Optimal configuration of  $p$ -cycles in WDM networks, in: IEEE International Conference on Communications - ICC, Vol. 5, 2002, pp. 2761–2765.

- [12] W. D. Grover, J. Doucette, Advances in optical network design with  $p$ -cycles: Joint optimization and pre-selection of candidate  $p$ -cycles, in: Proc. IEEE LEOS Summer Topical Meetings, 2002, pp. 49–50.
- [13] L. Ruan, F. Tang, Dynamic establishment of restorable connections using  $p$ -cycle protection in WDM networks, in: International Conference on Broadband Networks, Vol. 1, 2005, pp. 137–144.
- [14] B. Jaumard, C. Rocha, D. Baloukov, W. D. Grover, A column generation approach for design of networks using path-protecting  $p$ -cycles, in: Proceedings of IEEE/VDE Workshop on Design of Reliable Communication Networks - DRCN, 2007, pp. 1–8.
- [15] C. Rocha, B. Jaumard, P.-E. Bougué, Directed vs. undirected  $p$ -cycles and FIPP  $p$ -cycles, in: International Network Optimization Conference (INOC), 2009, pp. 1–9.
- [16] K. Murakami, H. Kim, Optimal capacity and flow assignment for self-healing ATM networks based on line and end-to-end restoration, IEEE/ACM Transactions on Networking 6 (2) (1998) 207–221.
- [17] V. Chvatal, Linear Programming, Freeman, 1983.
- [18] C. Barnhart, E. Johnson, G. Nemhauser, M. Savelsbergh, P. Vance, Branch-and-price: Column generation for solving huge integer programs, Operations Research 46 (3) (1998) 316–329.
- [19] IBM, IBM ILOG CPLEX 11.0 Optimization Studio (2011).
- [20] W. Grover, Mesh-Based Survivable Networks, Prentice-Hall, 2004.
- [21] D. Schupke, The tradeoff between the number of deployed  $p$ -cycles and the survivability to dual fiber duct failures, in: IEEE International Conference on Communications - ICC, Vol. 2, 2003, pp. 1428–1432.
- [22] R. Hülsermann, S. Bodamer, M. Barry, A. Betker, C. Gauger, M. Jäger, M. Köhn, J. Späth, A set of typical transport network scenarios for network modelling, Tech. rep., ITG Fachtagung Photonische Netze, Leipzig (2004).
- [23] P. Batchelor *et al.*, Ultra high-capacity optical transmission networks: Final report of action COST 239, Tech. rep., Faculty of Electrical Engineering and Computing, University of Zagreb (1999).
- [24] P. Cholda, A. Jajszczyk, Reliability assessment of optical  $p$ -cycles, IEEE/ACM Transactions on Networking 15 (6) (2007) 1579–1592.