Design of $p$-Cycles for full node protection in WDM Mesh Networks

Brigitte Jaumard *, Honghui Li *†

*CSE, Concordia University, Montreal, Qc, H3G 1M8, Canada, Email: bjaumard@cse.concordia.ca
† College of Computer and Information Engineering, Inner Mongolia Agricultural University, 010018, China

Abstract—We propose a $p$-cycle expanded protection scheme that can guarantee 100% node protection, in addition to 100% protection against single link failures. While some previous studies had already noted that $p$-cycles can naturally offer some node protection, we show that, at the expense of some $p$-cycle overlapping, with very mild impact on the bandwidth efficiency, we can guarantee node protection.

We propose a design and solution method based on large scale optimization tools, namely Column Generation (CG), which compute $p$-cycles offering both link and node protection. Previous models offer a solution where a large number of potential cycles needs first to be enumerated, leading to very large ILP models which cannot scale.

Comparisons are made between our proposed design approach and the work of Grover and Onguetou (2009). Results show that our approach clearly outperforms their design in terms of capacity efficiency and of the number of distinct cycles. We also show that our approach clearly outperforms their design in terms of node protection.

Index Terms—Survivable WDM networks, $p$-cycles, node protection, column generation.

I. INTRODUCTION

WDM techniques enable a single fiber to carry multiple non-overlapping channels, and a single channel can operate at the speed of up to 100 Gbps. WDM networks carry huge amounts of traffic. As optical WDM networks are occasionally prone to a single failure of network infrastructure (e.g., a fiber cut or a node failure), survivability is an essential requirement in the design of WDM networks.

Various types of protection approaches have been proposed for providing the survivability of WDM mesh networks. Among these approaches, the $p$-cycle (Pre-configured Protection Cycle) approach [1] holds the unique characteristic of ring-like restoration speed and mesh-like capacity efficiency. The last decade has seen numerous research studies on the design of $p$-cycles. The overwhelming majority of these studies explored the protection of WDM networks against a single link failure and did not worry about single node failures.

However, the failure of a single node may still occur due to some disasters such as fires or flooding taking down an entire node. A single node failure is equivalent to the failure of all its incident links. The consequences of a node failure are therefore significant. In addition, the node ports are not protected due to node cost and space limitation. The port failure of a node disrupts multiple channels which usually cannot be handled by link protection approaches, e.g., conventional $p$-cycles.

Besides the path-segment-protecting $p$-cycles [2] and path-protecting $p$-cycles [3] approaches which, with the use of $p$-cycles, resemble the segment and path protection schemes, few studies have investigated node protection with $p$-cycles. Stamatelakis and Grover [4] proposed node-encircling $p$-cycles (NEPCs), which are rather spare bandwidth consuming. Schupke [5] proposed and evaluated an Automatic Protection Switching (APS) protocol enhancement to provide means for node failure with $p$-cycles. Onguetou and Grover [6] proposed a new insight of node protection, which, with the help of overlapping segment $p$-cycles, either spanning or lying on the $p$-cycles, allow partial or full protection of all the intermediate nodes of the routing paths. The same authors in [7] later restrict the segments to two hop segments in order to retain the simplicity of $p$-cycle switching operations and to keep the ILP models easier to solve.

The conventional $p$-cycle design method formulates the design problem as an Integer Linear Program (ILP). There are two approaches for solving the resulting ILP. In the first one, it involves the off-line generation of either the whole set of potential $p$-cycles [1], [8], or a restricted set of promising candidate $p$-cycles [9], [10], leading either to a huge ILP or to a heuristic solution with unknown accuracy. The second approach relies on an implicit enumeration of the set of cycles thanks to the column generation techniques [11], and jointly generates and provisions the $p$-cycles.

In this paper, we investigate the design of $p$-cycles with 100% node protection. The underlying idea comes from the observation that a node is protected if its two adjacent links on the working path are supported by two on cycle links belonging to the same $p$-cycle. It resembles the two hop approach of Grover and Onguetou [7], with one additional feature, to be discussed in Section II-B.

The rest of the paper is organized as follows. In Section II, we explain the node protection scheme that we propose and its difference with the two hop approach of Grover and Onguetou [7]. In Section III, we present an efficient large scale optimization mathematical model for the design of $p$-cycles ensuring full node protection. Numerical results are presented in Section IV, where we compare our design with conventional $p$-cycles and with the method of Grover and Onguetou [7].
Conclusions are drawn in Section V.

II. NODE AND LINK PROTECTION p-CYCLES

We introduce the concept of overlapping p-cycles in order to ensure 100% node protection. It generalizes the node protection proposed in [7] in two respects: firstly, the node protection is embedded in the generation and the protection provisioning of the p-cycles and does not require a second optimization step once the (link) p-cycles have been selected, and secondly, p-cycles can handle the node protection of nodes even if they are crossed by several paths as long as those paths require independent protection from the p-cycle. As we will see in the numerical results, this allows reducing the spare bandwidth requirement.

A. Node protection in overlapping p-cycles

The overlapping p-cycle concept is alike the concept of overlapping segment protection. Indeed, while segment protection has been introduced as a compromise between link and path protection, there are two types of segment protection, the regular segment protection when protection segments have the same endpoints as the working segment, and the overlapped segment protection where segments overlap in order to guarantee node protection in addition to link protection, see Jaumard et al. [12]. We investigate here the overlapping of p-cycles in order to guarantee 100% node protection.

An illustration is provided in Figure I. 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 link or a straddling link of $C_1$ or $C_2$. Each intermediate node is also protected as its two adjacent links on the working path are protected by the same p-cycle, e.g., node $v_7$ has its two adjacent links $\{v_7, v_9\}$ and $\{v_7, v_{11}\}$ protected by $C_1$.

B. Node intersecting paths

The 2-hop strategy [7] only allows a p-cycle to protect a node with respect to only one affected working path. We allow the node protection of a node lying on several paths if the paths require disjoint protection for that node protection. Fig. 2(a) illustrates the idea. A network topology with 5 nodes is shown in Fig. 2(a), with three demands routed on routing paths $w_1$, $w_2$ and $w_3$, respectively. Fig. 2(b) shows the solution from the design strategy proposed in [7]. Therein, three p-cycles, $c_1$ and $c_2$ and $c_3$ are required to provide full node protection. The resulting spare capacity usage is nine channel units. However, with our approach, one p-cycle $c_4$ suffices to provide full node protection. Upon the failure of node $A$, on-cycle links $B-C, C-D$ and $D-E$ 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 five units of channels. Thus, compared to the 2-hop strategy of [7], our solution cuts 44.44% spare capacity usage.

III. A COLUMN GENERATION MODEL

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 $\omega_\ell$ be the number of traffic units 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 an optimization method based on a column generation (CG) technique for the design of p-cycles for full node protection using the overlapping p-cycle strategy described in Section II. The objective is to minimize the spare capacity usage such that 100% guaranteed survivability can be ensured with respect to the single failure of either a link or a node.

Following a CG modeling, 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 generates new p-cycles which improves, one iteration after the other one, the current value of the objective of the (continuous relaxation of the) master problem.

A. The master problem

The master problem relies on the concept of configurations where a p-cycle configuration $c$ is made of a one unit cycle, and the set of links and nodes protected by that cycle. Note that each node protection corresponds to a pair made of a node $v$ and a working path $p$ where $v$ is an intermediate node of $p$.

A p-cycle configuration $c$ is represented by a vector $(a_{\ell}^p)_{\ell \in L}$ and a matrix $(a_{vp}^p)_{p \in P, v \in V_p}$. The vector component $a_{\ell}^p \in [1, \cdots, d_p]$ suffices to provide full node protection using the overlapping p-cycle strategy.
\{2,1,0\} denotes the number of protection paths provided by the \(p\)-cycle \(c\) for the protection of link \(\ell\). The element \(a^c_{\ell} \in \{1,0\}\) and is equal to 1 if \(p\)-cycle \(c\) provides a backup path in order to overcome a node failure of \(v\) with respect to the working path \(p\). Let \(\text{COST}^c\) be the spare cost of the \(p\)-cycle \(c\).

Variables \(z^c\) denotes the number of copies of \(p\)-cycle configuration \(c\) that are selected in the protection scheme associated with the solution.

The mathematical model can then be written as follows.

\[
\min \sum_{c \in C} \text{COST}^c z^c
\]

subject to:

\[
\sum_{c \in C} a^c_{\ell} z^c \geq \omega_\ell \quad \ell \in L
\]  

(1)

\[
\sum_{c \in C} a^c_{pv} z^c \geq d_p \quad p \in P, v \in V_p
\]  

(2)

\[
z^c \in \mathbb{Z}^+ \quad c \in C
\]  

(3)

Constraints (1) ensure that the overall traffic is protected against a single link failure. Constraints (2) ensure that all demands are protected against a single node failure at a node \(v\) lying on working path \(p\), for all intermediate nodes on all working paths. Constraints (3) are variable domain constraints.

### B. The pricing problem

The goal of the pricing problem is to generate a promising \(p\)-cycle that would decrease the value of the current solution of the master problem. The pricing problem corresponds to the optimization problem with the objective of minimizing the so-called reduced cost of the master problem subject to a set of constraints for the generation of a \(p\)-cycle and for its set of protected links/nodes. The objective function can be written as follows.

\[
\min \sum_{\ell \in L} \text{COST}^c - \sum_{\ell \in L} u_\ell a^c_{\ell} - \sum_{p \in P} \sum_{v \in V_p} u_{pv} a^c_{pv}
\]

where \(u_\ell\) and \(u_{pv}\) are dual variables associated with constraints (1) and (2) respectively. \(\text{COST}^c\) is the spare capacity cost of \(p\)-cycle configuration \(c\), which is defined as the sum of the spare cost \(\Lambda_\ell\) of its on-cycle links.

Before setting the mathematical model of the pricing problem, we need to introduce the following notations:

Sets

- \(\omega(v)\) the set of links adjacent to node \(v\).
- \(\delta(v)\) the set of nodes adjacent to node \(v\).
- \(V_p\) the set of intermediate nodes on working path \(p\).
- \(P_v\) the set of working paths going through node \(v\).

Variables

- \(b_\ell = 1\) if link \(\ell\) is on the current cycle, 0 otherwise.
- \(s_\ell = 1\) if link \(\ell\) straddles the current cycle, 0 otherwise.
- \(y_v = 1\) if node \(v\) is on the current cycle, 0 otherwise.
- \(x^f_{pv} = 1\) if link \(\ell\) is used to protect a working path \(p\) against the failure of its intermediate node \(v\), 0 otherwise.

With these notations, the objective function of the pricing problem can be written as follows:

\[
\min \sum_{\ell \in L} \Lambda_\ell b_\ell - \sum_{\ell \in L} u_\ell (b_\ell + 2s_\ell) - \sum_{p \in P} \sum_{v \in V_p} u_{pv} \sum_{\ell \in \omega(\ell_{pv})} x^f_{pv}
\]

The pricing problem includes the 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 of constraints takes care of determining the pair \((p, v)\) made of a working path \(p\) and one of its intermediate node \(v\) such that \(v\) is protected by the cycle.

The first group of constraints is defined below.

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

(4)

\[
s_\ell \leq y_v - b_\ell \quad v \in V, \ell \in \omega(v)
\]  

(5)

\[
s_\ell \geq y_v + y_{v'} - b_\ell - 1 \quad v, v' \in V, \ell = \{v, v'\} \in L
\]  

(6)

Each node on a given cycle must have two incident links on the cycle. This is ensured by constraints (4). Constraints (5) and (6) are used to identify straddling links. These two sets of constraints say that a link can be a straddling link if its two end nodes are on-cycle and the link itself is not. Constraints (7) prevent generating a \(p\)-cycle which includes multiple cycles. Otherwise, it burdens identifying straddling links.

The second part of the constraints is next presented.

\[
x^f_{pv} \leq b_\ell \quad p \in P, v \in V_p, \ell \in L
\]  

(8)

\[
x^f_{pv} = 0 \quad p \in P, \ell \in \omega(v), v \in V_p
\]  

(9)

\[
\sum_{\ell \in \omega(v)} x^f_{pv} = \sum_{\ell \in \omega(\{v_1, v_2\})} x^f_{pv} \quad p \in P, v \in V_p
\]  

(10)

\[
\sum_{\ell \in \omega(\ell') \mid \ell \neq \ell'} x^f_{pv} \leq 2 \quad p \in P, v \in V_p, v' \in V
\]  

(11)

\[
x^f_{pv} \geq x^f_{p'} \quad p \in P, v \in V_p, \ell' \in \omega(\ell')
\]  

(12)

\[
\sum_{p \in P_v} x^f_{pv} \leq 1 \quad v' \in V \setminus \{v_1, v_2\}
\]  

(13)

\[
b_\ell, s_\ell, y_v, x^f_{pv} \in \{0,1\} \quad v \in V, p \in P
\]  

(14)

Constraints (8) ensure that only on-cycle links are eligible for protecting the working path \(p\) against a single failure of its relay node. Constraints (9) say that, if a link is adjacent to the relay node \(v\) of the working path \(p\), the link cannot be used by the associated protection paths. Constraints (10) - (12) are flow conservation constraints for defining the associated protection paths. Constraints (10) say that a protection path
must end at two end nodes of the associated 2-hop segment defined by working path \( p \) and its relay node \( v \). Constraints (11) and (12) together ensure that, for the nodes except the two end nodes of the associated 2-hop segments, the number of outgoing and incoming flows must be identical. Upon the failure of a node, working paths passing through the node are all disrupted. Constraints (13) say that a link channel can only be used for recovering one unit disrupted 2-hop segment. The final set of constraints contains variable domain constraints.

### IV. Computational Results

In this section, we evaluate the solution performances of our proposed CG-based design (NMpCycle) for full node protection. As the goal of this paper is to propose a scalable and capacity-efficient design method, we compare NMpCycle with GOpCycle proposed in [7] in terms of capacity redundancy. The capacity redundancy is defined as the ratio of spare capacity usage over working capacity usage [13]. Also, we compare capacity redundancy between \( p \)-cycles for node protection derived from NMpCycle with the conventional ones which only guarantee 100% link protection (LKpCycle) [11].

In addition, we evaluate the associated dual link failure restoration ratio of the \( p \)-cycles derived from GOpCycle, NMpCycle and LKpCycle, respectively. The dual link failure recovery ratio \( (R_{2}) \) is calculated as the total number of recovered traffic units over all dual link failure scenarios divided by the overall total number of dual failure affected traffic link pairs. For link-protecting \( p \)-cycles, \( R_{2} \) is calculated as in [14]. For node-protecting \( p \)-cycles, \( R_{2} \) is calculated in a similar way as for conventional link \( p \)-cycles except that a dual-link failure can be recovered if the two failed links are adjacent on the same working path.

GOpCycle, NMpCycle and LKpCycle were all implemented in C++ and were solved by CPLEX 11.0.1 MIP solver. The solutions of these three designs were obtained with an optimality gap less than 1.0%.

#### A. Data instances

We carry out experiments on five network instances for evaluation and comparison. Table I presents the network instances and their associated topology characteristics. For each network, we provide the number of nodes, the number of links, and the average nodal degree (as an indicator for the network connectivity). Moreover, we give the number of demand requests and working capacity usage (the number of link wavelength channels) for each traffic instance. Each demand request correspond to a one unit demand between a given node pair. It is a random value generated with a uniform random distribution on the interval [1..20]. Each demand request is routed along a shortest path.

#### B. Capacity redundancy

Fig. 3(a) depicts the comparisons of capacity redundancy of the three designs (GOpCycle, NMpCycle and LKpCycle) over five network instances. For each network instance, the \( p \)-cycles for node protection obtained from GOpCycle are more capacity redundant (less capacity efficient) than those from NMpCycle, keeping in line with the example in Section II-B. The redundancy differences between GOpCycle and NMpCycle vary from \( \sim 7\% \) to \( \sim 20\% \).

In addition, we can observe that the \( p \)-cycles for node protection have a higher capacity redundancy than those for the conventional link protection \( p \)-cycles. However, using the NMpCycle design, only marginal extra spare capacity is required for full link and node protection compared to that for link protection, which ranges from \( \sim 1\% \) to \( \sim 13\% \).

#### C. Dual link failure restoration ratio

Fig. 3(b) shows, for five network instances, the associated dual link failure restoration ratio \((R_{2})\) of the \( p \)-cycle sets obtained from GOpCycle, NMpCycle and LKpCycle, respectively. In general, \( R_{2} \) of the \( p \)-cycle set from NMpCycle is greater than \( R_{2} \) for LKpCycle while it is smaller than \( R_{2} \) for GOpCycle. This comes from the fact that GOpCycle is the most capacity redundant among these three designs. The \( R_{2} \) differences between NMpCycle and LKpCycle range from \( \sim 1\% \) to \( \sim 8\% \). For the NSF instance, the \( p \)-cycles from NMpCycle can achieve \( \sim 7\% \) more \( R_{2} \) than those from LKpCycle while only requiring no more than \( \sim 2\% \) extra redundant capacity.

#### D. Number and length of distinct cycles

In Fig. 3(c) (resp. Fig. 3(d)), we present the number (resp. the length) of distinct cycles in the optimal solutions of GOpCycle, NMpCycle and LKpCycle, respectively. For each network instance, LKpCycle requires the smallest number of distinct cycles among these three design methods. The differences between LKpCycle and GOpCycle (or NMpCycle) range from \( \sim 37\% \) to \( \sim 83\% \). The advantage of link-protecting \( p \)-cycles in terms of management is therefore considerable over node-protecting \( p \)-cycles. Regarding the \( p \)-cycle design methods for node protection, NMpCycle provides protection with the smallest number of distinct \( p \)-cycles than GOpCycle while with a similar number in the GERMANY and COST239 instances. The differences range from \( \sim 13\% \) to \( \sim 27\% \). Thereby, in terms of management, NMpCycle outperforms GOpCycle. Except for the Bellcore and the Germany instances, the length of the node protection \( p \)-cycles is smaller than the length of the node protection \( p \)-cycles of [7], and it is always smaller or equal to the length of the conventional \( p \)-cycles.

<table>
<thead>
<tr>
<th>Networks</th>
<th>Nodes</th>
<th>Edges</th>
<th>Node Degree</th>
<th>Demands</th>
<th>Working Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>GERMANY [15]</td>
<td>17</td>
<td>26</td>
<td>3.1</td>
<td>136</td>
<td>4050</td>
</tr>
<tr>
<td>BELLCORE [2]</td>
<td>15</td>
<td>28</td>
<td>3.7</td>
<td>105</td>
<td>2610</td>
</tr>
<tr>
<td>NJ LATA [16]</td>
<td>11</td>
<td>23</td>
<td>4.2</td>
<td>55</td>
<td>943</td>
</tr>
<tr>
<td>COST239 [17]</td>
<td>11</td>
<td>26</td>
<td>4.7</td>
<td>55</td>
<td>792</td>
</tr>
</tbody>
</table>
V. CONCLUSION

In this paper, we study the p-cycle design in WDM mesh networks for full link and node protection against a single failure. We showed that it is possible to modify the p-cycle design so as to ensure 100% node protection. Numerical results showed that ensuring node protection in addition to link protection against a single failure is not significantly more costly in terms of spare capacity than the link protection against a single failure.

REFERENCES


