A Simple Generalized Approach to Node Failure Recovery with Span-Protecting p-Cycles

Diane Prisca Onuetou and Wayne D. Grover
TRLabs, 1200 Harley Court, 10045-111 Street, T5K 2M5, Edmonton, AB, Canada
ECE Dept., University of Alberta, 2nd Floor ECERF, 9107-116 Street, T6G 2V4, Edmonton, AB, Canada
{donguetou, grover}@trlabs.ca

Abstract—This paper shows that, viewed in a generalized “two-hop” framework for node failure recovery, p-cycles actually have a very high inherent ability to restore paths transiting through a failed node. We also showed that with relatively little, if any, extra spare capacity, the principle is also amenable to explicit design of networks for 100% node and span failure protection with a single efficient set of p-cycles that support both functions. This is very different than the often-prevailing assumption that “ordinary” p-cycles offer no node protection, or only the same node-protecting property of a BLSR ring embodies. Indeed, the two-hop paradigm for recovery of affected paths transiting failed nodes could provide an attractive option for future network operators in that “ordinary” p-cycles are more localized, fast acting, and simple to plan and operate than any other option such as NEPCs, flow-protecting p-cycles or FIPPs.

Index Terms—ordinary p-cycles, two-hop, node failure protection, Rp_node

I. INTRODUCTION

Network survivability design is primarily focused on recovery from span failures because the frequencies of fiber cable cut events are hundreds to thousands of times higher than corresponding reports of transport layer node failures. Nevertheless much less frequent than span failures, node outages are particularly harmful when they arise, at least because each specific node failure involves the simultaneous failure of all node-incident edges. On the other hand, if optical-cross-connects (OXCs) tend to be highly robust and well protected, IP/MPLS routers still suffer downtimes about as frequently as span failures because of software patches, upgrades or even crashes. Thus, it is not utopian to consider protection against both span and node failures in the design.

Doing so, it has been known that end-to-end path-protecting architectures such as shared backup path protection (SBPP), demand-wise shared protection (DSP) and pre-cross-connected trails (PXT) inherently provide some protection against intermediate node failures arising somewhere along the working paths. Corresponding levels of node failure restorability depend on backup channel-capacities and node-disjointness considerations in the shared risk link groups (SRLGs). In contrast to path-oriented paradigms, span-protecting architectures are based on the deployment of a set of backup path-segments between the end-nodes (i.e., the “custodial” nodes) of a given failed span. Thus, integrating node failure recovery in span-protecting networks is much more challenging than with path-oriented protection.

If bidirectional-line-switched-rings (BLSRs) and unidirectional-path-switched rings (UPSRs) are recognized an inherent ability to recover paths transiting through a failed node, within the surviving portion of the ring, node failure protection using span-oriented paradigms in mesh-based survivable networks requires (in fact) some extensions of the original principle. A related illustration is the node-inclusive span survivability (NISs) scheme for span restoration in [2]. The key idea behind NISs is to define two custodial regions, one hop away from the custodial nodes with respect to each possible span failure. Doing so, a related, relatively small and localized instance of the path restoration problem can be solved for any failure affecting each given node-inclusive span entity (i.e., each span plus its custodial nodes).

p-Cycles are now a fairly known span-protecting scheme, with many interesting and attractive properties [3]-[5]. The original intention with span-protecting or “ordinary” p-cycles (as opposed to more recent FIPP p-cycles [13]-[18]) is efficient and fast protection against single span failures. A subsequent and common misunderstanding is that span-protecting p-cycles offer no form of node protection. More correctly, since inception, it has been realized that p-cycles do offer inherently the same protection to on-cycle paths traversing a failed node, as does a BLSR with respect to paths in the ring [6]-[7].

What has, however, remained less clear is how to protect paths that transit a node on a p-cycle and which have straddling relationship to the respective p-cycle. To protect those straddling paths against node failures as well, there have been various extensions to the basic node-protecting property of p-cycles. One main idea explored for node protection with p-cycles is the “node-encircling” principle studied and developed in, for example, ([6],[8]-[9]). Another line of work partly motivated by including node protection has lead to extensions of the whole p-cycle concept into path-segment or so-called “flow-protecting” p-cycles [10]-[11], and to end-to-end path protection with p-cycles [13]-[18].

Overall this contribution is extended from [1]. We explain and explore a two-hop flow strategy to node failure protection using ordinary p-cycles, which seems to have been overlooked to date. Section II presents the two-hop flow concept and compares this with prior related concepts. Section III formulates an integer-linear-programming (ILP) design model for the two-hop flow strategy, recalls equivalent ILP mathematical models for prior approaches, and proposes an adaptation of a novel combination of genetic-algorithms (GA) and ILP methods ([18]-[20]) to address large scale instances.
Section IV presents case studies, the test methodology and experimental results. Section V concludes the paper and indicates possible lines of future direction.

II. PROBLEM FORMULATION AND RELATED WORK

A. Two-Hop Flow: a New Insight and Approach to Node Failure Recovery using Ordinary p-Cycles

In brief, this contribution is to observe that the BLSR-like loopback reaction that ordinary p-cycles make to restore on-cycle flows transiting through a node is actually also applicable to straddling flows failing at a p-cycle node if the two spans adjacent to the failure node both end on another nodes on the same p-cycle. Of course, in hindsight, this is always true for on-cycle flows transiting a node; so this is a generalization of the prior known BLSR-like node protection condition. But the more general criteria can be seen to also allow protection of the following additional cases (with ordinary p-cycles) shown in Fig. 1.

Fig. 1 actually illustrates how an ordinary span-protecting p-cycle can restore any 2-hop path-segment intersecting the cycle structure upstream and downstream of a given failed node, whether or not the 2-hop flow-segment is entirely on the protecting structure. In Fig. 1(a), we have a p-cycle under the normal network state. In Fig. 1(b)-(e), we show how that p-cycle can be used to react in a previously overlooked way under node failure circumstances. The only requirement is that the end-nodes of the two-hop segment are on the same p-cycle as each other. So, Fig. 1(f) captures the class of situation where at least one end-node of the two hop-segment under consideration is not part of the cycle structure; this cannot be covered by the novel general criterion of node failure recovery.

In Fig. 1(b)-(e), the failure scenarios specifically consider whether the 2-hop segment is entirely on the protecting structure as in Fig. 1(b), or if the p-cycle crosses only one of the spans of the 2-hop segment as shown in Fig. 1(c), or if both spans of the 2-hop path-segment straddle the cycle as in Fig. 1(d), or if only the end-nodes of the 2-hop flow are part of the cycle structure as shown in Fig. 1(e). One way to think about this is to consider any 2-hop flow as a kind of “virtual span”: doing so the three cases in Fig. 1(b)-(d) are all equivalent to on-cycle (span) failures; while the situation in Fig. 1(e), in which neither of the two spans comprising the 2-hop segment nor the failed node are part of the cycle, corresponds to a p-cycle reacting to a straddling (span) failure.

The practical importance of the 2-hop standpoint is that the simplicity of operation of ordinary p-cycles is retained and only one set of span-protecting candidate structures can be employed in a complete design for both 100% span and node failure protection. Subsequently, the 2-hop strategy stands in contrast with prior attempts to protect against node failure events through concepts such as node-encircling p-cycles (NEPCs—[6],[8]-[9]), full path-segment (or more simply “flow”) protecting p-cycles [10]-[11], and failure-independent path-protecting (FIPP) p-cycles [13] to [18].

B. Two Hops versus Node-Encircling p-Cycles

A p-cycle is said to be an NEPC for a given “encircled” node if it contains all the neighbor-nodes of the encircled node, but not the given node itself. Thus the key property is that, an NEPC intercepts any flow transiting the encircled node, and hence (with suitable capacity) can reroute all affected transiting flows when the node fails. For example, the p-cycle in Fig. 2(a) is an NEPC for node G and, as shown in Fig. 2(b), intercepts every path transiting through the encircled node G. In contrast, the given p-cycle cannot be an NEPC for other vertices of the graph because nodes A, I, D, E, F and H are themselves part of the protecting structure; while off-cycle nodes B and C are neighbors each other and thus, B has its neighbor-nodes C (and C its neighbor node B) out of the cycle structure.

But, because an NEPC does not include (by definition) the protected node itself, this approach does not exploit the inherent reaction p-cycles can have against on-cycle node failures. Fairly often as well, some nodes may have no simple NEPCs in a graph, especially in sparser networks. Non-simple candidate cycles crossing a span or node more than once can be still considered, but this adds greatly to the operational and conceptual complexity. Designing a separate set of NEPCs generally also

---

**Legend:**
- p-Cycle structure
- Failed 2-hop segment
- Failed Node
- Protection segment

**Fig. 1** Intrinsic “Two-hop” Node Protecting Capabilities of p-Cycles

**Fig. 2** The Concept of NEPCs
requires significantly more spare capacity in the complete design because the NEPCs provide for node recovery separately from other p-cycles which are still needed for span protection.

The 2-hop strategy is more flexible than NEPCs. By considering failed-segments of (only) two hops, we eliminate the node-encircling constraint. Instead, for every potential node failure scenario, we select a subset of ordinary cycles that cover (if considered together) all the neighborhood of the failed node—rather than looking for NEPCs of the failed node. Results to follow show much greater overall capacity efficiency as well.

C. Two Hops versus Flow-Protecting p-Cycles

In other prior work, full path-segment (or flow) protecting p-cycles can also support node failure recovery. The principle is to observe that every p-cycle will also happen to intersect a number of working flows upstream and downstream. Subsequently, any intermediate node or span failure along each respective intersecting flow-segment can be restored within the p-cycle, exactly as with the conventional switching mechanism. For instance, Fig. 3(a) shows two cycles X and Y respectively intersecting path-segments [A,C] and [B,D] of a given working flow. In Fig. 3(b), cycle X offers two restoration routes in the event of span or node failure along segment [A,C]. This is also the case for cycle Y and the segment-portion [B,F]; but as shown in Fig. 3(c), there is only one protection route within cycle Y if the span/node failure occurs along the portion [F,D].

Flow-protecting p-cycles generalize the 2-hop strategy in the sense that protected segments may freely go from one or more spans to entire working paths. But by restricting failed flows to be considered and restored strictly only as if they were two-hop segments, we only require the same simple and local type of failure detection and pre-defined switching plans as for span failures. To illustrate, the situation in Fig. 3(c) requires advanced inter-nodal signaling or centralized management to activate the right restoration actions, which are different depending on where the failed path-segment is disrupted—i.e. whether along [B,F] or [F,D].

However, if the added complexity of a failure-dependent reaction is accepted, we recently showed that very high levels of node failure restorability could be achieved by applying a path-segment view to ordinary span-protecting p-cycles [12].

D. Two Hops vis-à-vis Failure-Independent Path-Protecting p-Cycles

Although this requires switching from span- to path-oriented paradigms, failure-independent path-protecting (FIPP) p-cycles with proper node-disjointness constraints also stand as a valid alternative approach to node failure protection using p-cycles. FIPP p-cycles actually operate like conventional p-cycles but they are chosen so that each protects a set of end-to-end paths that are mutually span- (and when desired, node-) failure disjoint between end-nodes on the FIPP structure [13]-[14]. Literature indicates disjoint-route-sets (DRS) and column-generation (CG) as practical methods for FIPP network planning [15]-[17].

When FIPP failure independency constraint is relaxed so that a given working path can be assigned different p-cycles depending on where the failure occurs, the principle is referred as general path-protecting (GPP) p-cycles. It has been shown in prior research that optimal GPP solutions are very close to being FIPP solutions because in general, no more than 2 working paths remain unprotected after the constraint of failure independence is imposed onto GPP designs [18]. Thus as is the case in this paper, it is not awkward to consider GPPs as if they were FIPPs. The merit is that relative to FIPP p-cycles, GPPs can be very efficiently captured in a mathematical formulation.

III. MATHEMATICAL PROGRAMMING ASPECTS

The conventional p-cycle minimum spare capacity design model is given as starting point.

A. Conventional p-Cycle Minimum Spare Capacity Design Model

The following definitions serve for the p-cycle minimum spare capacity design model.

Sets:
- $S$ is the set of spans in the network, indexed by $i$ for failing spans and $j$ for surviving spans or spans in general.
- $P$ is the set of candidate cycles, determined by a pre-processing method and indexed by $p$.

Input Parameters:
- $C_j$ is the cost of each unit of capacity (i.e. channel) on span $j$.
- $w_j$ is the number of working channels to be protected on span $i$. This is an input arising from whatever routing process is employed for demand matrix.
- $x^p_j \in \{0,1,2\}$ encodes the number of restoration path-segments that a single unit-sized copy of $p$-
cycle \( p \) may provide to span \( i \). \( x_{ip}^p = 2 \) if \( i \) straddles \( p, x_{ip}^p = 1 \) if \( p \) crosses \( i \), and \( x_{ip}^p = 0 \) otherwise.

- **Decision Variables:**
  - \( s_j \) is the number of spare channels on span \( j \) in the design.
  - \( \eta^p \) is the number of unit-sized copies of \( p \)-cycle \( p \) in the design.

**ILP Formulation:**

Eq. (1) is to minimize total spare capacity requirements while Eq. (2) guarantees full protection in the event of single span failures, through the \( p \)-cycles built within spare capacity in Eq. (3).

\[
\text{Minimize } \sum_{j \in S} C_j \cdot s_j . \tag{1}
\]

Subject to:

\[
w_i \leq \sum_{p \in P} x_{ip}^p \cdot \eta^p , \forall i \in S . \tag{2}
\]

\[
s_j = \sum_{p \in P \eta^p = 1} \eta^p , \forall j \in S . \tag{3}
\]

**B. ILP Models for the 2-Hop Approach to Node Failure Protection using Ordinary \( p \)-Cycles**

The following additional definitions serve for the ILP maximizing \( R_{1,\text{node}} \) in \( p \)-cycle designs.

- **Additional Sets:**
  - \( N \) is the set of nodes in the network, indexed by \( k \).
  - \( D \) is the set of demands, indexed by \( r \). We assume all units for a given demand-pair \( r \) take the same working route.

- **Additional Input Parameters:**
  - \( d^r \) is the number of units of capacity for demand-pair \( r \).
  - \( \delta_k^r \in \{0,1\} \) encodes end-nodes for demand-pair \( r \). \( \delta_k^r = 1 \) if node \( k \) is either origin or destination of \( r \), \( \delta_k^r = 0 \) otherwise.
  - \( \varepsilon_k^r \in \{0,1\} \) indicates which nodes are on the working route of demand-pair \( r \); \( \varepsilon_k^r = 1 \) if \( r \) crosses \( k \) en route, and \( \varepsilon_k^r = 0 \) otherwise.
  - \( \mu_k^{p,r} \in \{0,1,2\} \) indicates how many protection routes are available within the cycle structure \( p \) to restore a 2-hop segment for demand-pair \( r \), which has at its end-nodes on \( p \)-cycle \( p \) and \( k \) as intermediate node: \( \mu_k^{p,r} = 2 \) if \( k \) is off-cycle; \( \mu_k^{p,r} = 1 \) if \( k \) is on-cycle; and \( \mu_k^{p,r} = 0 \) otherwise.

- **Additional Decision Variables:**
  - \( n_r^{p,r} \) is the number of unit-copies of \( p \)-cycle \( p \) allocated to demand-pair \( r \) in order to prevent node \( k \) failures.
  - \( \theta_k^{p,r} \) is the number of capacity units for demand-pair \( r \) effectively rerouted within \( p \)-cycle \( p \) when node \( k \) fails.

\[ \Lambda_k, T_k, \Theta_k \text{ record, in the event of node } k \text{ failure, statistics on affected, transiting and recovered traffic.} \]

\[ \text{The following inequality is always true: } \Lambda_k \geq T_k \geq \Theta_k . \]

**ILP Formulation:**

1) **Maximizing Node Failure Restorability Level in a Conventionally Designed \( p \)-Cycle Network**

An assumption to this first new ILP is to keep as is the routing of working paths, the spare capacity and the \( p \)-cycles selected in an otherwise conventional \( p \)-cycle minimum spare capacity design. And the objective is to maximize node failure restorability as stated in Eq. (4).

To calculate node failure recovery level—Eq. (5) allocates restoration path-segments, available within the selected \( p \)-cycles, to working paths transiting the intermediate node of any 2-hop segment. Doing so, Eq. (6) asserts that only intersecting flows are potentially restorable. Eq. (7) keeps the assignment of protection path-segments under the actual spare capacity of available \( p \)-cycles. And Eq. (8) gives no credit to potentially protected paths that would exceed the actual demand volume present.

Eqs. (9)-(11) are for statistics only: they respectively record the demand volume affected by a given node outage, the amount that is potentially restorable because transiting the failed node, and the number of working paths that are effectively protected in the design. Node failure restorability is given by

\[ R_{1,\text{node}} = \frac{\sum_{k \in N} \Theta_k}{\sum_{k \in N} \Lambda_k} . \tag{4} \]

Subject to:

\[ \theta_k^{p,r} \leq n_r^{p,r} \cdot \mu_k^{p,r} , \forall r \in D , \forall k \in N , \forall p \in P . \tag{5} \]

\[ n_r^{p,r} \leq \mu_k^{p,r} \cdot \infty , \forall r \in D , \forall k \in N , \forall p \in P . \tag{6} \]

\[ \eta^p \geq \sum_{p \in P} n_r^{p,r} , \forall k \in N , \forall p \in P : \varepsilon_k^r = 1 . \tag{7} \]

\[ \sum_{p \in P} \theta_k^{p,r} \leq d^r , \forall r \in D , \forall k \in N : \varepsilon_k^r = 1 \text{ and } \theta_k^0 = 0 . \tag{8} \]

\[ \Lambda_k = \sum_{r \in D} \varepsilon_k^r \cdot d^r , \forall k \in N . \tag{9} \]

\[ T_k = \sum_{r \in D, \delta_k^r = 0} \varepsilon_k^r \cdot d^r , \forall k \in N . \tag{10} \]

\[ \Theta_k = \sum_{r \in D, \delta_k^r = 0} \varepsilon_k^r \cdot \theta_k^{p,r} , \forall k \in N . \tag{11} \]

2) **\( R_{1,\text{node}} \) Maximization with Controlled or No Penalties over Minimum Spare Capacity Requirements**

Rather than maximizing node failure recovery in a pre-planned 100% span restorable \( p \)-cycle network, we can nudge the minimum spare capacity solution to happen to support simultaneously the maximum feasible level of node failure restorability. Setting a suitable small \( \varepsilon \), this is achievable with the bi-criteria objective in Eq. (12) and the constraints (2)-(3) and (5)-(11).
Minimize \( \sum_{j \in S} C_j \cdot s_j + \alpha \cdot \sum_{k \in N} (T_k - \Theta_k) \). \hfill (12)

One might want to assert, instead, maximum \( R_{1 \text{-node}} \) subject to an allowable extra budget \( \xi \) relative to the minimum spare capacity cost \( B \) for 100% restorability against single span failures. If so, Eq. (13) will be accordingly added to the prior set of constraints and Eq. (4) will be then considered as objective function.

\[
\sum_{j \in S} C_j \cdot s_j \leq B \cdot (1 + \xi). \hfill (13)
\]

In a different way, one can also ask how much spare capacity is required at minimum to guarantee 100% restorability against both node and span failures. The equivalent ILP is given by Eqs. (1)-(11), without the objective function in Eq. (4), and the new constraint (14) that transforms the inequality (8) in an equality satisfying the needs for full node failure recovery.

\[
\sum_{p \in \Phi} \Theta^p_k = \epsilon^p \cdot d^r, \quad \forall r \in D, \quad \forall k \in N : \delta^p_k = 0. \hfill (14)
\]

3) Multiple Quality of Protection Concerns

The above-defined mathematical formulations are easily adaptable to multiple-quality-of-protection (multi-QoP) purposes. Assume that a new input parameter \( \phi^p \in \{0,1,2\} \) identifies three service classes requiring: 

- 100% \( R_{1 \text{-node}} \) if \( \phi^p = 2 \), maximum possible \( R_{1 \text{-node}} \) within available \( p \)-cycles when \( \phi^p = 1 \), and no form of node failure protection for \( \phi^p = 0 \).

The multi-QoP design model is defined by the objective function in Eq. (12) and constraints (2)-(3), (5)-(11) and (14). Doing so, Eqs. (5)-(6) will now apply to the situations where \( \phi^p > 0 \); and Eqs. (8) and (14) will stand for \( \phi^p = 1 \) and \( \phi^p = 2 \), respectively. Given that node failure recovery levels are known to be 0 and 100% for the first and third service classes, statistics in Eqs. (9)-(11) will belong to \( \phi^p = 1 \) only.

C. Equivalent ILP Formulations for Prior Node Restoration Options with p-Cycles

1) Flow-Protecting p-Cycles and NEPCs

All prior-defined ILPs are applicable as is to the cases of flow-protecting \( p \)-cycles and NEPCs. The principle is just to recognize that each \( p \)-cycle has its own spans for which it provides the intended span failure protection but exploits and/or adjusts the design in a way that \( p \)-cycles also act as 2-hop protecting \( p \)-cycles, flow-protecting \( p \)-cycles or NEPCs when it comes to prevent node failures. From the flow-protecting \( p \)-cycle perspective, the parameter \( \mu^p_k \) has to be pre-processed for full path-segments (of one or more spans to entire working paths).

From the NEPC viewpoint, \( \mu^p_k = 2 \) for every \( r \) transiting \( k \) if and only if \( p \) is an NEPC for node \( k \); otherwise, \( \mu^p_k = 0 \).

On the other hand, because of the scarcity of NEPCs (especially in sparser networks), it might happen that all paths cannot topologically survive every single node failure. In such cases, 100% node (and span) restorable designs are not achievable. Thus, the multi-QoP ILP for NEPCs mudge maximum possible level of node failure restorability for \( \phi^p = 2 \) with the prior bi-criteria objective. Eq. (15) can serve as objective function, with suitable settings for \( \beta \) and \( \gamma \). And Eqs. (9)-(11) have to be adjusted in a way that \( \Lambda_k, T_k, \Theta_k \) records statistics.

\[
\min \sum_{k \in N} (T_{2k} - \Theta_{2k}) + \beta \cdot \sum_{j \in S} C_j \cdot s_j + \gamma \sum_{k \in N} (T_{1k} - \Theta_{1k}). \hfill (15)
\]

2) Path-Protecting p-Cycles

The following additional definitions serve in GPP related formulations.

**Additional Input Parameters:**

- \( \delta^p_j \in \{0,1\} \) indicates spans that demand-pair \( r \) crosses en route; \( \delta^p_j = 1 \) if \( r \) crosses span \( j \), and \( \delta^p_j = 0 \) otherwise.

- \( y_p \in \{0,1,2\} \) encodes the number of protection segments that one unit-sized copy of \( p \)-cycle \( p \) may provide to demand-pair \( r \). \( y_p = 2 \) if working route of \( r \) fully straddles \( p \)-cycle \( p \), \( y_p = 1 \) if \( r \) is in a full or partial on-cycle relationship with \( p \), and \( y_p = 0 \) otherwise.

**Additional Decision Variables:**

- \( m^p_r \) represents the number of copies of \( p \)-cycle \( p \) assigned to demand-pair \( r \) in the design.

**ILP Formulation:**

The conventional \( p \)-cycle minimum spare capacity design model is not applicable to GPP \( p \)-cycles, because the latter requires to switch from span- to path-based protection. To achieve full span restorable GPP designs, Eqs. (1) and (3) can be combined with the following constraints. Eq. (16) guarantees working paths to survive any single span failure. And Eq. (17) selects and capacitate cycle structures, providing enough cycle copies to handle rival working routes that share one or more spans in common.

\[
\sum_{p \in P} y_p^r \cdot m^p_r \geq d^r, \quad \forall r \in D. \hfill (16)
\]

\[
\sum_{r \in D} m^p_r \cdot \delta^p_i \leq \eta^p, \quad \forall i \in S, \quad \forall p \in P. \hfill (17)
\]

Regarding node failure protection purposes, Eq. (18) is an equivalent of Eq. (17) but for node-disjointness requirements. The assignment of protection segments within available \( p \)-cycles, previously done by Eqs. (5)-(6), is now achieved by Eq. (19). Eq. (20) calculates cycle copies required in the design, and all Eqs. (7)-(14) remain the same.

\[
\sum_{r \in D} m^p_k \cdot \delta^p_k \leq \eta^p, \quad \forall k \in N, \quad \forall p \in P. \hfill (18)
\]
\[ \theta_{p}^{r} \leq n_{r}^{p}, \quad \forall r \in D, \quad \forall k \in N, \quad \forall p \in P. \quad (19) \]
\[ n_{r}^{p} \leq m_{r}^{p}, \quad \forall r \in D, \quad \forall k \in N, \quad \forall p \in P. \quad (20) \]

Table 1 gives a summary ILP mathematical design models. All of them were implemented in AMPL 10.100 and solved using CPLEX 10.1.0 with a mixed integer programming gap for optimality (MIPGAP) of 10^{-4}, on an Intel Duo Core Processor running Mac OS X 10.5.8 at 2.8 GHz with 4 GB of 1067 MHz DDR3. Where ILP problem instances were solvable with the complete set of candidate cycles, the whole process (including preparatory programs) typically reached full termination in about 15 minutes or less.

We managed the size of large problem instances by restraining the number of eligible cycles. Our preselection technique relates to a novel combination of GA with ILP methods which seems to have many features to recommend it for any large \( p \)-cycle problem involving the selection of a relatively few optimal candidate cycles from an almost infinite space. Note that this added (only) a couple more minutes in running times for network instances under consideration in this paper.

D. A GA-ILP Heuristic to Solving Node-Protecting \( p \)-Cycle ILPs at Very Large Scale

The GA-ILP preselection concept follows the normal steps of a GA-like evolutionary heuristic. The initial set of candidates is first partitioned into subsets of equal sizes, each subset comprising an individual of which genome corresponds to the index numbers of the cycles constituting the subset in question. The union of the above individuals then embodies an initial population for the GA-ILP, and the node-protecting \( p \)-cycle problem under consideration is solved using in turn each of the individuals. Every individual is subsequently assigned a weight equivalent to the objective function value of the constituent ILP and the \( n/2 \) individual-pairs showing the optimum sum of weights are selected for breeding. Every selected pair of individuals produces two children by crossing the first half of one parent’s genome with the second half of the second individual’s genome and vice versa. To maintain genetic diversity into the offspring, there is a specific mutation policy that consists of randomly substituting cycle indexes in some children for solution cycles of the individuals not selected to reproduce. The generational process of evaluation, selection, crossover and mutation is repeated on new populations, until the objective function value of the constituent ILP does not improve anymore from one individual to another. All unique cycles of the last population then comprise the reduced set of candidates that is still, to our knowledge, \( O(\text{individual size}) \).

Whereas the space of all possible candidates is fully enumerable, the GA-ILP solution to a conventional \( p \)-cycle network design problem is expected to be equivalent to what would be obtained if the instance under consideration was solved with the entire set of candidate cycles. As a form of test, in work presented at [21] we showed that even for almost 85,000 candidate structures, the GA-ILP always reaches optimality for problem instances to which exact (ILP) solutions are known. But the purpose for GA-ILP is to go onto much larger problem sizes where enumeration is not practically possible. In this regard, we can actually recognize [21] two additional problem classes based on whether the space of all possible candidate cycles is enumerable but impractical to import into the ILP solver, or not even enumerable in practice. The GA-ILP provided a high quality solution for an instance of about 387,740 candidates. In fact, that solution was found equal (within the MIPGAP being employed) to what was obtained using the CG approach, meaning that the GA-ILP was still within 1% of optimality. In other prior work, we successfully applied the GA-ILP framework for \( p \)-cycle network designs with controlled optical path length in the restored network state [19], for near-optimal FIPI \( p \)-cycle network designs through GPP [18], and to a 200-node challenge case which represents a specific instance of the third class of problems [20].

IV. EXPERIMENTAL RESULTS AND DISCUSSION

A. Case Studies

Five test case networks, shown in Fig. 4, were used. The first group of columns in Table 2 gives their number of nodes \( N \), spans \( S \), demand-pairs \( D \) and eligible cycles \( |P| \). “Havana” in Fig. 4(a) is a previously used network [22]. Two sets of demands are considered for it: the original matrix of 58 demand-pairs with units distributed on the interval [0.5], and another traffic matrix assuming connection requests between every single pair of nodes, with volumes uniformly distributed.

<table>
<thead>
<tr>
<th>Model</th>
<th>Problem Description</th>
<th>ILP Formulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Conventional reference ILP. ( p )-cycle minimum spare capacity.</td>
<td>Reference values and actual cycles for use in model 2—Eqs. (1)-(3).</td>
</tr>
<tr>
<td>2</td>
<td>Maximum ( R_{\text{node}} ), given a 100% span restorable design as input.</td>
<td>Assuming an existing set of p-cycles and working paths—Eqs. (4)-(11).</td>
</tr>
<tr>
<td>3</td>
<td>( p )-Cycle minimum spare capacity planning under ( R_{\text{node}} ) maximization.</td>
<td>Biriteria minimization of capacity and node unrecoverability—Eqs. (12), (2)-(3), (5)-(11).</td>
</tr>
<tr>
<td>4</td>
<td>( R_{\text{node}} ) maximization with controlled penalties over minimum capacity.</td>
<td>Merge models 1 and 2 plus extra budget—Eqs. (2)-(11), (13).</td>
</tr>
<tr>
<td>5</td>
<td>Full protection against both span and node failures.</td>
<td>Capacity minimization under full ( R_{\text{node}} )—Eqs. (1)-(3), (5)-(7), (9)-(11), (14).</td>
</tr>
<tr>
<td>6</td>
<td>Multi-QoP services for ( R_{\text{node}} )</td>
<td>Merge models 3 and 5—Eqs. (2)-(3), (5)-(12), (14).</td>
</tr>
<tr>
<td>7</td>
<td>Service differentiation for NEPCs.</td>
<td>Model 6 for NEPCs—Eqs. (2)-(3), (5)-(11), (15).</td>
</tr>
<tr>
<td>8</td>
<td>GPP ( p )-cycle minimum spare capacity.</td>
<td>Model 1 for GPP—Eqs. (1)-(3), (4)-(16)-(17).</td>
</tr>
<tr>
<td>9-13**</td>
<td>Similar to models 2-6, but for GPP.</td>
<td>Eqs involved—(1), (3)-(4), (7)-(14), (16)-(20).</td>
</tr>
</tbody>
</table>

*Although in varc, eqs. for ILPs 9 to 13 are picked up similarly to models 2 to 6.
hop-count based routing is applied to the original demand
There is one more Havana network instance in which a
18,335.1 for Bellcore; and 246,375 for Euro network.

kms of 23,934 for Havana when considering the original
another European network has 323 demand-pairs

(i) Initial routing

(ii) Other routing

(iii) Larger traffic matrix

Cost239 (shortest distance routing)

Italy (shortest distance routing)

Bellcore (shortest distance routing)

Euro (shortest distance routing)

on the interval [1..100]. The 2nd network is the well-
known Cost239 pan-European network, in Fig. 4(b).
Related traffic matrix includes 55 non-zero demand-pairs
uniformly distributed on [1..10]. The other test case
instances are given in Fig. 4(c)-(e): the Italy network has
78 demand-pairs distributed on [1..10], the Bellcore
network has 104 demand-pairs distributed on [1..20],
and another European network has 323 demand-pairs
distributed on the interval [1..2].

A shortest distance based routing is applied under most
normal network states, resulting in total working channel-
kms of 23,934 for Havana when considering the original
set of demands; 2,595,800 for Havana using the large
traffic matrix; 137,170 for Cost 239; 62,232 for Italy;
18,335.1 for Bellcore; and 246,375 for Euro network.
There is one more Havana network instance in which a
hop-count based routing is applied to the original demand
matrix; this leads to 166 working channels. The 3rd and
4th columns report statistics on paths affected by potential
node failure events and the number of such paths that can
be considered for restoration—the ratio of transiting over
total paths (including terminating paths which cannot be
restored) varies from 25 to 60%.

Six types of results are considered here to assess
effectiveness of the proposed “two-hop” node recovery
principle. In a first set of experiments corresponds to
models 1 and 8, in which nothing special is done for node
failure recovery in the design. These are test cases
designed for R1-span= 1 at minimum spare capacity.
Within each network, we then use the 2nd and 9th ILPs to
simulate each node failure and experimentally determine
what is the best R1-node level that can be obtained through
the two-hop and other comparative node recovery
methods. Further types of result obtained with models 3
and 10 show the level of R1-node that is achievable “for
free” under each principle (i.e., with no investment
beyond that needed only for R1-span= 1; but free to bias
the solution towards choosing cycles that also increase R1-node
levels). With enhanced ILPs 4 and 11, we maximize node
failure protection under given spare capacity budgets.
The 5th and 12th ILP-based results are to see how much
spare capacity has to be added to strictly assert 100% R1-
ode by each method being compared. The final type of
results relates to service differentiation and is achievable
with the 6th, 7th and 13th ILPs.

B. Performance of the 2-Hop Strategy

The 2nd column of Table 3 gives conventional p-cycle
minimum spare capacity design solutions for each of the
networks under consideration; these are 100% span
restorable only, with no node failure concerns. The 3rd
and 4th columns in Table 3 characterize node failure
restorability against both single span and node failures.
With enhanced ILPs 4 and 11, we maximize node
failure protection under given spare capacity budgets.
The 5th and 12th ILP-based results are to see how much
spare capacity has to be added to strictly assert 100% R1-
ode by each method being compared. The final type of
results relates to service differentiation and is achievable
with the 6th, 7th and 13th ILPs.

Table 2 Test Case Networks Characteristics

<table>
<thead>
<tr>
<th>Networks</th>
<th>[N]</th>
<th>[S]</th>
<th>[D]</th>
<th>[P]</th>
<th>A</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>Havana</td>
<td>17</td>
<td>26</td>
<td>58</td>
<td>135</td>
<td>32</td>
<td>4</td>
</tr>
</tbody>
</table>
| (i) Initial routing | Shortest distance based routing → 23,934 working channel-kms. | 271 | 77% i.e. 28% of affected
| (ii) Other routing | Least hop-count based routing → 166 working units. | 263 | 69% i.e. 26% of affected
| (iii) Larger  | Traffic matrix | 136(D) uniformly distributed | 29,317 | 14,591 | 50% of affected
| Cost239        | (shortest distance routing) | - Total of 137,170 working channel-kms. | 471 | 119% i.e. 25% of affected
| (shortest distance routing) | - Total of 137,170 working channel-kms. | 1357 | 521% i.e. 38% of affected
| (shortest distance routing) | - Total of 18,335.1 working channel-kms. | 1326 | 396% i.e. 30% of affected
| Italy           | 13   | 24   | 78   | 557  | 359 | 11 |
| Bellcore        | 15   | 28   | 104  | 976  | 847 | 8   |
| Euro            | 32   | 42   | 323  | 699  | 458 | 9   |

Fig. 4 Five Test Case Networks

(a) Havana: 17 nodes, 26 spans, 135 candidates
(b) Cost239: 11 nodes, 26 spans, 3531 candidates
(c) Italy: 13 nodes, 24 spans, 557 candidates
(d) Bellcore: 15 nodes, 28 spans, 976 candidates
(e) Euro: 32 nodes, 42 spans, 699 candidates
Overall, very high levels of $R_{1\text{-node}}$, typically 77% to 96%, are achieved under min-costs. Moreover, with additional capacity penalties of 0.89 to 21% we can achieve full restorability. Overall, a total of 74 paths (out of 77) survive single node failure conditions, for a very high level of up to 96% $R_{1\text{-node}}$. And this can be pushed to 100%, using the 5th ILP for assertion of 100% $R_{1\text{-node}}$, with less than 1% of additional spare capacity requirements. (The conventional $p$-cycle minimum spare capacity design solution for Havana involves 16 channel-copies of 4 distinct $p$-cycle structures, for spare capacity requirements of 20,264 channel-kms corresponding to 84.56% of redundancy to total distance-weighted working capacity.)

### C. Comparison to Prior Related Approaches

Let us now compare the 2-hop approach to the three other approaches; i.e. NEPC, flow-protecting $p$-cycles and GPP, for which results are given by 2nd, 3rd and 4th sets of data in the 3rd and 4th columns of Table 3. Noticeably the 2-hops approach is nearly as capacity-efficient as path-segment protecting $p$-cycles, with a difference of 0 to 20% for both metrics under observation. With 2-hops, flow-segments are otherwise shortened to guarantee, in addition, a straightforward failure detection and a real-time activation of right restoration processes.

NEPCs show $R_{1\text{-node}}$ is almost non-existent under min-costs; furthermore, it is usually not topologically possible to achieve full $R_{1\text{-node}}$, and it too costly when possible as with the Cost239 network. Consider again the Havana network as a supporting example. Table 2 indicates 32 NEPCs over the 135 distinct simple candidates available across the network graph. Those NEPCs cover only four nodes, i.e., Berlin, Hamburg, Hanover and Norden.

### Table 3 Sample Results on ILPs 1-4 and 8-11

<table>
<thead>
<tr>
<th>Networks</th>
<th>Conventional Design (spare capacity, redundance)</th>
<th>Surviving Paths under Min. Capa. Design, Level of $R_{1\text{-node}}$</th>
<th>Min. Capa. for 100% (or max) $R_{1\text{-node}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Havana</td>
<td></td>
<td>2-hop</td>
<td>nepc</td>
</tr>
<tr>
<td>(i) Initial routing</td>
<td>20,264 i.e. 85% (gpp: 20,451 i.e. $\Delta=+0.92%$)</td>
<td>74 i.e.</td>
<td>0 i.e.</td>
</tr>
<tr>
<td>(ii) Other routing</td>
<td>134 i.e. 81% (gpp: 157 i.e. $\Delta=+17.16%$)</td>
<td>51 i.e.</td>
<td>0</td>
</tr>
<tr>
<td>(iii) Larger traffic matrix</td>
<td>3,187,827 i.e. 1237 (gpp: 3,174,532 i.e. $\Delta=-0.42%$)</td>
<td>12,799 i.e. 87%</td>
<td>0</td>
</tr>
</tbody>
</table>

© 2010 ACADEMY PUBLISHER
among them, sole Hanover was carrying (about 11) transiting paths in the node payload characterization previously shown in Fig. 5. The 11 working paths thus correspond to the maximum achievable $R_{1\text{-node}}$ of 14%, which requires about 39% of extra spare capacity over min-costs. The usage of non-simple cycles is necessary to achieve an $R_{1\text{-node}}$ of 100%. In contrast to NEPCs, the path-segment strategy greatly improves on both maximum $R_{1\text{-node}}$ under min-costs and spare capacity penalties to reach 100% $R_{1\text{-node}}$. (Another bench of comparison, based on the 4th ILP, is given in Fig. 6 for Havana and Cost239 network; y-axis maximizes $R_{1\text{-node}}$ under extra spare capacity indicated in x-axis).

GPP involves shifting from span to path protection, so the related minimum spare capacity design solutions differ from that of conventional span-protecting p-cycles (otherwise used as benchmark for 2-hops, flows and NEPCs). Comparative min-cost solutions are both given in the 2nd column of Table 3. Overall, results seem to suggest that FIPP gives rise to lowest costs for instance cases involving more candidate cycles while conventional p-cycles are more adapted to smaller candidate spaces. For example, the Havana minimum capacity GPP-design solution is 100% $R_{1\text{-node}}$; there is however an indirect penalty as GPP minimum spare capacity design (i.e. 20,451 channel-kms) is more expensive than that of ordinary p-cycles (i.e. 20,264 channel-kms). Furthermore, in these test networks at least, GPP min-cost requirements are even higher than what is required to reach full node restorability using either flows (i.e., 20,335) or 2-hops (i.e., 20,444).

### D. Performance under Multi-QoP Requirements

Overall, the 2-hop strategy shows such a great performance that it seems to be a promising mixed priority service environments. Table 4 records multi-QoP experimental results for Havana network with the original traffic matrix and the shortest distance routing. As shown in the first row, four scenario types were considered: no node failure protection, maximum failure protection under minimum spare capacity requirements, 100% node failure restorability and several mixed scenarios. To distribute traffic among service classes for the scenario (50, 30, 20), for example, we randomly generated a number on the interval [1..100], considering in turn each demand-pair. Paths of the respective demand were then considered from class 2 (i.e. $\phi^r = 2$) if the generated number was on the interval [1..50], class 1 if on [51..80] and class 0 if in [81..100].

Results in the 2nd row in Table 4 indicates same capacities as in the conventional minimum spare capacity design for almost all multi-QoP scenarios. Up to 50% of traffic flows can be offered full node failure recovery while maintaining more than 90% of $R_{1\text{-node}}$ for 3/5th of the remaining demands, with exactly zero penalty over

---

### Table 4 Sample Results on Multi-QoP

<table>
<thead>
<tr>
<th>scenarios</th>
<th>(0,0,100)</th>
<th>(100,0)</th>
<th>(0,100)</th>
<th>(50,50,100)</th>
<th>(50,100,50)</th>
<th>(100,50,50)</th>
<th>(50,50,20)</th>
<th>(50,20,50)</th>
<th>(20,50,50)</th>
<th>(20,50,50)</th>
</tr>
</thead>
<tbody>
<tr>
<td>spare capacity</td>
<td>20,264</td>
<td>20,264</td>
<td>20,444</td>
<td>20,264</td>
<td>20,264</td>
<td>20,264</td>
<td>20,264</td>
<td>20,264</td>
<td>20,264</td>
<td>20,264</td>
</tr>
<tr>
<td>redundancy 84.66%</td>
<td>84.66%</td>
<td>84.66%</td>
<td>85.42%</td>
<td>84.66%</td>
<td>84.66%</td>
<td>84.66%</td>
<td>84.66%</td>
<td>84.66%</td>
<td>84.66%</td>
<td>84.66%</td>
</tr>
<tr>
<td>affected</td>
<td>(0, 0, 271)</td>
<td>(0, 271, 0)</td>
<td>(271, 0, 0)</td>
<td>(141,104,26)</td>
<td>(141,65,65)</td>
<td>(52,154,65)</td>
<td>(76,169,26)</td>
<td>(76,65,136)</td>
<td>(52,89,136)</td>
<td>(52,89,136)</td>
</tr>
<tr>
<td>transiting</td>
<td>(0, 0, 77)</td>
<td>(0, 77, 0)</td>
<td>(77, 0, 0)</td>
<td>(39,28,10)</td>
<td>(39,21,17)</td>
<td>(12,48,17)</td>
<td>(22,45,10)</td>
<td>(22,17,38)</td>
<td>(12,27,38)</td>
<td>(12,27,38)</td>
</tr>
<tr>
<td>protected</td>
<td>(0, 0, n/a)</td>
<td>(0, 74, n/a)</td>
<td>(77, 0, n/a)</td>
<td>(39,26, n/a)</td>
<td>(39,19, n/a)</td>
<td>(12,46, n/a)</td>
<td>(22,43, n/a)</td>
<td>(22,17, n/a)</td>
<td>(12,27, n/a)</td>
<td>(12,27, n/a)</td>
</tr>
<tr>
<td>$R_{1\text{-node}}$ for $\phi^r = 1$</td>
<td>n/a</td>
<td>96.10%</td>
<td>n/a</td>
<td>92.85%</td>
<td>90.47%</td>
<td>95.83%</td>
<td>95.56%</td>
<td>100%</td>
<td>100%</td>
<td></td>
</tr>
</tbody>
</table>
what is required for 100% R_{1-span} designs. On the other hand, more than 96% of R_{1-node} is achievable if all demands are considered from class 1 (i.e. \( p^o = 1 \)). R_{1-node} = 1 is possible for every traffic flow with less than 1% of extra spare capacity. (The 3rd, 4th and 5th rows give statistics on affected, transiting and protected paths in the event of any node failure.)

E. Effectiveness of the GA-ILP

In experiments conducted, ILP models 3 to 7 were too large for practical solution using the complete set of candidate cycles in Cost239 and Euro graphs. And we faced this problem more often with GPP ILPs 9 to 13, even when conventionally designed Cost239, Bellcore and Euro networks. Large problem instances were addressed within the GA-ILP framework in Section III.D. Fig. 7 shows the GA-ILP convergence for the 2-hop variant of model 5, i.e. 100% R_{1-node}. In related experiments, partitions of the complete space of candidates correspond to n=6 individuals for Havana, 40 for Cost239, 20 for Italy and Euro networks. The generational process typically completed earlier before the 20th iteration. For Havana, Italy and Bellcore networks of which exact solutions were known, the GA-ILP solution was always within 1% of optimality. Thus, we trusted the GA-ILP solutions for the Cost239 and Euro networks. (We reached the same conclusions for Cost239, Euro and Bellcore when substituting the constituent p-cycle model Euro for flow, NEPC and GPP approaches.)

V. Conclusion

Recent work has revealed a new, relatively simple and possibly cost-effective approach to achieve combined protection of optical networks against both node and span failures. The new principle is based on a generalization of how nodes in a BLSR-ring or p-cycle (to date) derive survivability through loopback at the nearest two neighbor-nodes on the same ring. The generalization views any combination of node failure and an affected transiting path from the standpoint of the 2-hop segment defined by the failure node, and the nodes immediately adjacent on the affected path. We then ask whether these nodes are found together within the same p-cycle as the failure node, or another p-cycle entirely. In any case where they are, we show that the transiting path affected by the node failure is inherently restorable by ordinary p-cycle switching actions whether the respective two-hop segment is on-cycle, straddling, or partially on-cycle and partially straddling. The novel combination of GA-methods with ILP was adapted for node-protecting p-cycles through 2-hop, flow and NEPCs.

The resulting network designs use only a single set of p-cycle structures that have the same or only slightly more capacity than a corresponding optimal set of p-cycles for span protection “only”. In this paper, we explained the principle and characterize its effectiveness in terms of network-wide single node failure restorability (R_{1-node}) in networks designed only for minimum spare capacity, networks designed for enhanced R_{1-node} (at min capacity) and networks designed strictly for R_{1-node}^{opt}. As well, the proposed approach for node-protecting p-cycles is compared to related prior concepts. A subsequent line of future direction is to develop 2-hop protecting p-cycles towards no more distinction between node and span failures.

REFERENCES


Fig. 7 GA-ILP Solutions for Full R_{1-node} with 2-Hops


Diane Prisca Onguetou is a PhD candidate in Electrical and Computer Engineering department at TRLabs and the University of Alberta, under the direct supervision of Professor Wayne Grover. She successfully completed in 2005 a M.Sc. in Telecommunications at INRS, Montréal (Québec), Canada. In 2002, she had a special postgraduate training in Signal Processing at "Université de Mame-La-Vallée" in France, and in 2001 she obtained an Electrical and Telecommunications from "Ecole Polytechnique de Yaoundé", Cameroon.

Her PhD research project seeks the goal of increasing the awareness and understanding of p-cycle design methods, through an ongoing series of advanced research in the design of p-cycle survivable networks. Findings to date include an elegant control of restored state path lengths, a cost-effective generalization of how nodes in a BLSR-ring or p-cycles derive survivability through loopback nearest two neighbor nodes on the same ring, whole fiber switched p-cycles and a novel combination of ILP and GA methods to solve the large scale p-cycle design problem. Her PhD defense is expected in Summer 2010.

Wayne D. Grover holds a B.Eng. (EE) from Carleton University, Ottawa; an M.Sc.(EE Science) from the University of Essex, England; and a Ph.D. (EE) from the University of Alberta. Dr. Grover has issued patents on 26 topics each issued in several countries, 76 journal publications, six book chapters and over 100 technical reports, seminars, and conference papers. In August 2003 his book Mesh-based Survivable Networks: Options and Strategies for Optical, MPLS, SONET and ATM Networking was published by Prentice-Hall (841 pages plus web-based appendices).

Three of his research papers, and his Ph.D. thesis on Self-Healing Networks, have become "highly cited" in different technical areas but he is most widely recognized for work in restorable network design and operation, including Sonet, ATM, DWDM and IP/MPLS networks. Following his decade-long development and advocacy of the concepts of self-healing and self-organizing transport networks, he is considered as a founding inventor in this field. In 1999 he received the IEEE Baker Prize Paper Award for his paper “Self-organizing Broadband transport networks” in the Oct. 1997 IEEE Proceedings.

Other research contributions are in the areas of high-speed synchronization, precise time transfer, wireless traffic analysis, rate-adaptive subscriber loops, radio-location in wireless systems and availability analysis of transport networks. Dr. Grover was an NSERC E.W.R. Steacie Fellow for 2001-2002. Previously he was the McCalla Professor in Engineering and recipient of the Martha Cook-Piper Research Prize (both at U of A.), the "Smart City" Award (City of Edmonton) and a Technology Commercialization Award from TRLabs (1997) for the licensing of technology to industry. In 2002 he was made an IEEE Fellow ("for contributions to survivable and self-organizing broadband transport networks.") and Fellow of the Engineering Institute of Canada. In 2003 he is serving as General Chair for the 4th International Workshop on Design of Reliable Networks (DRCN 2003), Banff, Alberta, Oct. 19-22,2003.