Reprint of paper to appear at the *2004 International Conference on Communications (ICC 2004)*, Paris, France, June 2004.

Strategies for Enhanced Dual Failure Restorability with Static or Reconfigurable *p*-Cycle Networks

Dominic A. Schupke*
Siemens AG, Corporate Technology
Information and Communications
Otto-Hahn-Ring 6, 81730 Munich, Germany
E-mail: <firstname>.<lastname>@siemens.com

Wayne D. Grover, Matthieu Clouqueur
TRLabs and University of Alberta
7th Floor, 9107 116 St NW
Edmonton, Alberta, Canada T6G 2V4
E-mail: {grover, clouqueur}@trlabs.ca

***Abstract*—We suggest methods for achieving high dual-failure restorability in *p*-cycle networks that are optimally designed only to withstand all single failures, or have minimized amounts of additional capacity for dual failure considerations. In one set of circumstances we consider static *p*-cycles (that cannot be rearranged once established) and propose strategies to enhance the dual-failure restorability based on concepts of failure spreading and limiting the maximum number of protection relationships of any *p*-cycle. We then consider the case of reconfigurable *p*-cycles, in which the spare capacity can be reconfigured dynamically, creating a new set of *p*-cycles that are optimized to withstand possible second failures. Results show significant improvements of the dual-failure restorability obtained by limiting the number of protection relationships in static design. For 100% dual-failure restorability, *p*-cycle reconfiguration reusing most of the existing *p*-cycles and adding some new ones appears to be the most promising approach.**

Keywords- p-Cycle, Protection, Dual Failures, Reconfiguration.

I. Background

The concept of *p*-cycle protection finds application in many kinds of networks [1]. High capacity-efficiency and fast protection switching times are achieved for span-protection in transport networks [1][2]. Like a shared-protection ring, a *p*-cycle protects working capacity on the spans it covers ("on-cycle spans") by providing loop-back protection paths. But unlike rings, a *p*-cycle can also protect working capacity of off-cycle spans which have their end-points on the *p*-cycle ("straddling spans"). Moreover, a *p*-cycle offers two protection paths for each straddling span, and, thus, protects twice as much working capacity on these spans as on the on-cycle spans. For a detailed description of the *p*-cycle concept see [1] or [3].

Most of the work on *p*-cycle network design to date has focused on efficiently providing a guarantee of 100% restorability against any single span failure. This "$R_1 = 1$" condition remains the most cost-effective single step that network operators can take to enhance the availability of all services in their networks. Once this is in place, however, it is reasonable to consider how the investment in redundant capacity to achieve $R_1 = 1$ might be further leveraged or optimized to also withstand dual failures as well as possible. Especially in large networks [4][5], or where maintenance activities reduce the available spare capacity, dual failures may become the dominant factor in determining the ultimate service availability of those networks. We therefore investigate several ideas for providing networks, based on *p*-cycles, which have assured survivability to any single failure, but also have an enhanced or optimized level of dual-failure restorability, denoted R_2. These strategies may involve using only the spare capacity already invested for $R_1 = 1$, or adding additional capacity to achieve specific goals for R_2.

Restorability is defined as the fraction of demands that are affected by a failure scenario, but which survive by virtue of recovery mechanisms using spare capacity of the network. For single span failure scenarios, *p*-cycles guarantee (by the usual design default condition) restorability R_1 of 100%. Some initial analyses of dual-failure restorability with static *p*-cycles have already been published [6][7]. This paper further extends and refines the methods for static *p*-cycle networks with enhanced dual-failure restorability and includes completely new methods for reconfigurable *p*-cycle networks that achieve 100% dual-failure restorability. In general, we can identify three reasons to trigger dynamic *p*-cycle reconfiguration [8]:

1. The demand changes: *p*-Cycles adapt to protect newly arrived demands [9][10].
2. A (first) failure occurs: Immediate *p*-cycle reconfiguration can maximize the potential for protection against a subsequent failure.
3. Operator/protocol-initiation: *p*-Cycles are re-optimized, e.g., after failure-repair or in a maintenance window.

The paper focuses on case 2; the related aspects of case 3, however, will be discussed.

For simplicity, we assume that in a dual failure scenario, the second failure always occurs after the first failure has been fully responded to by *p*-cycle protection. In other words any second failure is assumed to occur on a network in which the first failure and any reconfigurations in response to it have been completed. The investigations on static *p*-cycles [6] show that the dual-failure restorability can be improved by increasing the

*This work originated during a research stay of D. Schupke at TRLabs, Edmonton, Canada and is part of the work when he has been with the Institute of Communication Networks at Technische Universität München, Munich, Germany.

diversity of cycles used in the design and by designing so as to manage the number of protection relationships each p-cycle may have through a "susceptibility" measure. In this paper, we investigate these and other methods for both static and reconfigurable p-cycles, to achieve the highest state of readiness after a first failure should a second failure occur. In particular, static and reconfigurable p-cycles are compared with respect to capacity requirements and we consider whether the number of reconfiguration actions following a first failure can be minimized while still achieving goals for R_2. In Section II we classify the design cases arising when dual failures are taken into account and formulate corresponding optimization models. Section III discusses restorability and capacity-efficiency results for network case studies designed using the proposals of Section II. Section IV concludes and gives an outlook on further research.

II. P-CYCLE DESIGN MODELS TO ACHIEVE DUAL-FAILURE RESTORABILITY

Figure 1 gives a logical classification of design and operational approaches that can be considered from a standpoint of R_2 considerations. The options depend on the network operation (e.g., if reconfiguration is supported or not) and on design preferences (e.g., design for single or dual-failure restorability). Dual-failure optimization (R_2 optimization) means that the design aims to achieve high dual-failure restorability, with respect to the average case (or, alternatively the worst case).

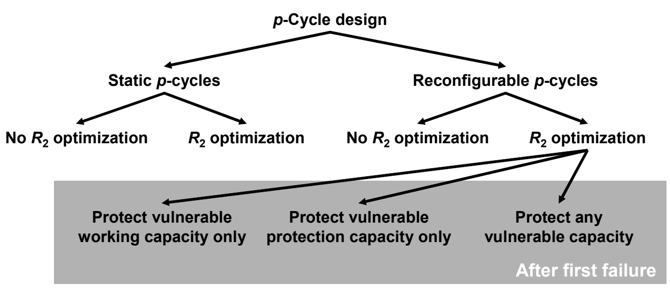

Figure 1. Possibilities in the p-cycle network design when dual failures are taken into account.

Let us now discuss design formulations corresponding to the different options in Figure 1.

A. Basic Models for p-Cycles Without R_2 Optimization

This formulation, first given in [9] and employed in [1][2], serves as the benchmark for R_1 design (single failure restorable) at minimum spare capacity cost and is also the starting point for the new models developed here. It assumes WDM networks with full (or sufficient) wavelength conversion, or networks with equivalent characteristics. The topology is described by the set of spans $\boldsymbol{S}$, indexed by i for failed spans and by k for non-failed spans. A set of eligible and simple[1] cycles $\boldsymbol{X}$ is pre-computed, including the protection capacity matrix, (π_k^x), indicating on which spans $k \in \boldsymbol{S}$ each candidate p-cycle $x \in \boldsymbol{X}$ lies, and the working capacity coverage matrix, or matrix of protection relationships, (ρ_i^x), indicating how many working capacity units on span $i \in \boldsymbol{S}$ can be protected by a single p-cycle on cycle $x \in \boldsymbol{X}$ (1 for on-cycle spans, 2 for straddling spans, 0 otherwise). For a span $k \in \boldsymbol{S}$, the cost of one unit of capacity is denoted by C_k. The design solution is described by variables n^x for the number of unit copies of p-cycles selected on cycle $x \in \boldsymbol{X}$, and by auxiliary variables s_k for the number of units of protection (or "spare") capacity on span k. We assume the working capacities w_k are given input parameters.

$$\text{Minimize: } \sum_{k \in \boldsymbol{S}} C_k \cdot s_k \quad (1)$$

$$s_k = \sum_{x \in \boldsymbol{X}} \pi_k^x \cdot n^x \;, \;\forall k \in \boldsymbol{S} \quad (2)$$

$$w_i \le \sum_{x \in \boldsymbol{X}} \rho_i^x \cdot n^x \;, \;\forall i \in \boldsymbol{S} \quad (3)$$

$$n^x \in \mathbb{N} \;, \;\forall x \in \boldsymbol{X}$$

The objective function (1) minimizes the total cost-weighted protection capacity, constraints (2) determine the protection capacity needed to form the p-cycles and constraints (3) ensure that there are enough p-cycles to protect the working capacity.

B. Models for Static p-Cycles with R_2 Optimization

1) Reduce Susceptibility to Dual Failures

Reference [6] proposes the following extension to the basic model. It aims to reduce the susceptibility[2] by keeping the working capacity coverage (or protection relationships) of the cycles low. New variables t^x indicate whether a cycle is used in the design, i.e., t^x is 1 if n^x is greater than 0, otherwise it is 0:

$$\frac{n^x}{N} \le t^x \le n^x \;, \;\forall x \in \boldsymbol{X} \quad (4)$$

$$t^x \in \{0,1\} \;, \;\forall x \in \boldsymbol{X}$$

The constant N is a large number. The additional (minimax) variable $\rho_{\max}$ is minimized (this can be included in the objective, see [6]):

$$\rho_{\max} \ge \sum_{i \in S} \rho_i^x \cdot t^x \;, \;\forall x \in \boldsymbol{X} \quad (5)$$

By minimizing $\rho_{\max}$ we limit the maximum number of unit-capacity protection relationships provided by any single p-cycle. This improves the dual-failure restorability by reducing the propensity for any single p-cycle to be simultaneously affected by two failures. New results here, show that a different susceptibility measure, σ_i^x, relating to the number of protected spans instead of capacity units, allows an even better control of the dual-failure restorability.

[1] A simple cycle passes through a node or a span at most once.

[2] A p-cycle is said to be *susceptible* to a specific dual failure combination if it bears protection relationships to working capacity on two or more spans which are both failed simultaneously in the failure scenario.

We define σ_i^x as:

$$\sigma_i^x = \frac{\rho_i^x + \pi_i^x}{2} \quad (6)$$

And (5) is replaced by:

$$\sigma_{\max} \geq \sum_{i \in \boldsymbol{S}} \sigma_i^x \cdot t^x \ , \ \forall x \in \boldsymbol{X} \quad (7)$$

The matrix entry σ_i^x indicates that cycle *x* can protect capacity on span *i*, i.e., σ_i^x = 1, if span *i* bears an on-cycle or a straddling span logical relationship to cycle *x*, 0 otherwise. This is without consideration of the actual amount of working capacity that cycle *x* may protect on span *i*. Hence, a cycle *x* is susceptible to $\sum_{i \in \boldsymbol{S}} \sigma_i^x \left(\sum_{i \in \boldsymbol{S}} \sigma_i^x - 1 \right)$ dual failures.

Now, instead of including the susceptibility in the objective, we can take it directly into account during the selection of the eligible *p*-cycles. We thereby incorporate susceptibility as a restriction, i.e., a candidate *p*-cycle in $\boldsymbol{X}$ is responsible for at most $\sigma_{\max}$ spans.

2) Failure Dispersal or Fault Sharing

Another principle is to design so that the total protection of working capacity on any given span is diversified or dispersed over multiple *p*-cycles. The intent is to avoid the situation where a dual failure takes down the entire working demand going through a span because they have all been protected by the same *p*-cycle which may be fully utilized by the first failure or otherwise overwhelmed by the particular dual failure combination. A first approach is to require that at least μ diverse cycles share the protection relationships on a span (this is similar to the "multiplicity" or sharing factor concept for a different kind of shared backup cycles in [11] where a multiple protection cycles per span are required)[3]:

$$\sum_{x \in \boldsymbol{X}} \sigma_i^x \cdot t^x \geq \mu \ , \ \forall i \in \boldsymbol{S} \quad (8)$$

For instance, with reference to Figure 2, if μ is 2, then the span 7-2 can be protected by *p*-cycle combinations A/B, A/C, B/C, or A/B/C. Note that after a first failure of this span, the combination A/B is better than A/C, since the cycles A and B are disjoint except for this span.

If the two corresponding cycles each cover half of the working capacity of span 2-7, then the worst case second failure for combination A/B (e.g., 2-3) causes loss of at most half of the working capacity, while the worst case second failure for combination A/C (e.g., 2-3) causes the full working capacity to be lost.

[3] This constraint can only hold for spans that are members of at least μ cycles. Although we can reformulate to $\sum_{x \in \boldsymbol{X}} \sigma_i^x \cdot t^x \geq \mu$, $\forall i \in \boldsymbol{S} : \sum_{x \in \boldsymbol{X}} \sigma_i^x \geq \mu$, for simplicity we assume μ to be low enough, such that every span is member of at least μ cycles.

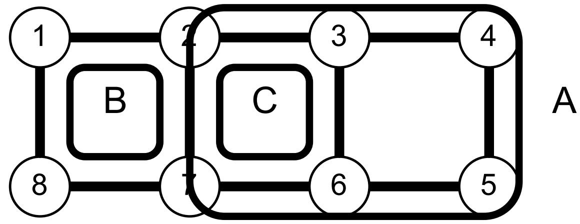

Figure 2. Example of the fault-sharing concept.

C. Models for Reconfigurable p-Cycles with R_2 Optimization

After a first failure, *p*-cycles that are used for restoration of affected working links are no longer available to protect other unaffected working links that they normally protect. In addition, the protection path(s) of *p*-cycles which are in an active restoral state are themselves exposed, should a second failure hit them directly. For instance, the failure of span 2-7 in Figure 3 makes working capacity on spans 2-3, 3-4, 4-5, 5-6, 6-7, and 3-6 vulnerable to a second failure, if *p*-cycle A protects some working capacity on these spans. Also the protection path (2-3, 3-4, 4-5, 5-6, and 6-7) is vulnerable.

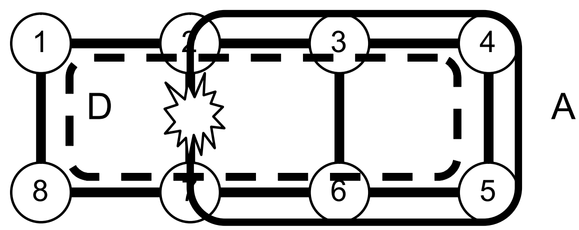

Figure 3. Example of the *p*-cycle reconfiguration concept.

p-Cycle reconfiguration can "hedge" against this vulnerability by introducing, for example, cycle D after the failure. To fully protect the working capacity and the protection path, two copies of cycle D are needed.

If the *p*-cycle is not perfectly loaded, it may not be necessary for it to be able to protect all spans within itself. In the example, *p*-cycle A may be responsible only to protect span 2-7; then, *p*-cycle D in the reconfiguration needs (for dual-failure protection) the same capacity as cycle A only, since it protects the protection path only. To avoid making the reconfiguration models that generalize these ideas overly complex, we make the following assumptions for a surviving span *k* after a first failure on span *i*:

1. Reconfiguration should protect an upper bound on the unprotected working capacity following the first failure, i.e., it should protect $w_{k,i}^{'} = \sum_{x \in \boldsymbol{X},\, \rho_i^x > 0} \rho_k^x \cdot n^x$ units on span *k*. This term sums the total number of working capacity units on span *k* that have potentially lost their protection because that protection was guaranteed by *p*-cycles also protecting span *i*. This number is therefore greater than or equal to the working capacity that is actually left unprotected after failure of span *i*. Equality holds only for perfectly loaded *p*-cycles on span *k*. Overall, this assumption can result in more capacity consumption than needed, because reconfiguration protects the vulnerable

working capacity of perfectly loaded p-cycles. If, however, the p-cycles are well loaded, little additional capacity will be expected.

2. Reconfiguration should protect an upper bound on the capacity of the unprotected protection paths that are activated by the first failure, i.e., it should protect $s'_{k,i} = \sum_{x \in \boldsymbol{X},\, \rho_i^x > 0} \pi_k^x \cdot n^x$ units on span k. This term sums the total protection capacity on span k of all p-cycles affected by the first failure on span i. Note that, again, this assumption can result in more capacity consumption than needed, because reconfiguration will protect a p-cycle's protection capacity even if it is not actually used after a first failure on one of its on-cycle or straddling spans (i.e., the first failed span could have been but was not actually protected by the p-cycle). Note that a perfectly loaded p-cycle protects all its on-cycle and straddling spans (i.e., the upper bound is reached).

So far we have also considered that both the exposed (but not failed) working capacity following the first failure, and the protection path(s) themselves should be considered for re-establishing protection coverage during reconfiguration in response to a second failure. Either of them, however, can in principle also be considered alone. If a working path is guaranteed to survive a single failure only, it may be sufficient to regard the aim of the post-failure reconfiguration as being to achieve protection of the remaining unfailed working capacity only. In other words, the philosophy would be to protect normal working capacity from dual failures, but not to reprotect protection paths themselves. This points out that there can actually be two slightly different meanings to what one has in mind when we say "dual-failure survivability." The central distinction is whether:

- Every working demand should be protected from both failures.
- Every working demand should be protected from any one failure that hits it directly, even if the network as a whole has already sustained a prior failure.

These two contexts are considered in our results as "R_2 = 100%" and "R_2 = 100% only for working capacity," respectively. As pointed out in [7], certain dual failures can be survived with failure signalling on the cycle or under certain failure circumstances, but for the reasons given this is not considered in this section. In addition, we assume that existing working capacity and the routes of working demands that are unaffected by a first (or second) failure are not re-arranged.

1) Complete Cycle Reconfiguration After First Failure

Complete reconfiguration can be thought of as meaning that for each single failure state in the network we have a predetermined optimized set of alternate p-cycles to switch to, to maximize the protection against a subsequent failure. In addition the no-failure state has a specific configuration of p-cycles that is known to support 100% restorability to any first-failure.

To model this we introduce an additional set of variables denoting which p-cycles are to be formed after failure of span i:

$$n_i^x \in \mathbb{N},\ \forall x \in \boldsymbol{X},\ \forall i \in \boldsymbol{S}:\ \pi_i^x = 0$$

Note that for certain dual failures (e.g., in a ring-like p-cycle without straddling spans) reconfiguration cannot achieve 100% dual-failure survivability. Then, for these dual failures n_i^x is not defined by default (i.e., by $\pi_i^x = 0$). Constraints (2) are replaced by constraints determining the upper limit of spare capacity required for each failure case:

$$s_k \ge \sum_{x \in \boldsymbol{X}} \pi_k^x \cdot n^x,\ \forall k \in \boldsymbol{S} \tag{9}$$

$$s_k \ge \sum_{x \in \boldsymbol{X},\, \pi_i^x = 0} \pi_k^x \cdot n_i^x,\ \forall k,i \in \boldsymbol{S},\ k \ne i \tag{10}$$

These additional constraints determine the set of cycles after failure of span i:

$$w_k + \sum_{x \in \boldsymbol{X},\, \rho_i^x > 0} \pi_k^x \cdot n^x \le \sum_{x \in \boldsymbol{X},\, \pi_i^x = 0} \rho_k^x \cdot n_i^x,\ \forall k,i \in \boldsymbol{S},\ k \ne i \tag{11}$$

The second term in (11) sums the capacity of the protection paths after the first failure. Overall, the solution specifies the set of p-cycles to use to fully withstand any second failure after each first-failure state has been entered. It also generates enough spare capacity on each span so that all of the various p-cycle configurations are feasible to configure.

Reconfiguration can be done in practice using cross-connect nodes (XC) with pre-planned information: once the failure-span identity is disseminated network-wide, each XC independently implements its local set of spare capacity connections that effects the next required p-cycle set network wide. Since the p-cycle configuration is pre-planned, its computation can be done offline, and thus, is not a time-critical factor. Alternately, networks with the distributed cycle pre-configuration (DCPC) protocol [9] can re-activate the protocol to autonomously self-organize the required new set of p-cycles after any single-failure state is entered (and dissolving any p-cycles not used in the first failure protection response).

2) Incremental Configuration After a First Failure

Another approach is to increment the existing configuration in each single-failure state. In other words, after a first failure, we want the p-cycles, which are initially found and which are partially active in recovering the first failure, to stay. Therefore, to protect against second failures, we deploy additional p-cycles and, as an option, assign more (previously unused) protection relationships to the existing p-cycles.

The most straightforward incremental approach involves introducing only an additional set of p-cycles. These additional variables denote how many additional cycles are employed in the reconfigured state after failure of span i:

$$\Delta n_i^x \in \mathbb{N},\ \forall x \in \boldsymbol{X},\ \forall i \in \boldsymbol{S}:\ \pi_i^x = 0$$

The constraints (2) are replaced by constraints determining spare capacity needed in the non-failure case plus the upper limit of spare capacity for each failure case:

$$s_k \geq \sum_{x \in \boldsymbol{X}} \pi_k^x \cdot n^x + \sum_{x \in \boldsymbol{X},\, \pi_i^x = 0} \pi_k^x \cdot \Delta n_i^x \ , \ \forall k,i \in \boldsymbol{S} \ , \ k \neq i \qquad (12)$$

The following additional constraints determine the set of additional cycles after failure of span i:

$$\sum_{x \in \boldsymbol{X},\, \rho_i^x > 0} \rho_k^x \cdot n^x + \sum_{x \in \boldsymbol{X},\, \rho_i^x > 0} \pi_k^x \cdot n^x \leq \sum_{x \in \boldsymbol{X},\, \pi_i^x = 0} \rho_k^x \cdot \Delta n_i^x \ , \quad \forall k,i \in \boldsymbol{S} \ , \ k \neq i \qquad (13)$$

With this modification of the model, additional *p*-cycles are found and the initial *p*-cycles remain untouched in their configuration (they do not even get new protection relationships). As the initial *p*-cycle set and thus a large part of the network configuration is kept autonomous, this may ease administration of the protection configuration.

However, to realize better capacity efficiency, we can also exploit the initial set of cycles to protect against secondary failures. In this approach, besides deploying new *p*-cycles, existing *p*-cycles get new protection relationships. In this case we can use the complete cycle reconfiguration approach of the previous subsection plus the following constraints:

$$n^x \leq n_i^x \ , \ \forall x \in \boldsymbol{X} \ , \ \forall i \in \boldsymbol{S} : \ \pi_i^x = 0 \qquad (14)$$

3) *Minimizing Reconfiguration Actions*

The number of cross-connections to effect these reconfigurations could be a concern. Theoretically a complete update of protection configuration may be needed. Depending on technology and control assumptions this may or may not be an issue. Reconfiguration is assumed to happen after the first failure is fully protected and before a second failure has occurred. It can therefore be a relatively slow process (minutes) and still be very short relative to the repair time of the first failure or the mean time to a second failure. If the nodes are able to do parallel operations (e.g., parallel cross-connecting instructions over the connection control interface), such concerns may also not be of importance. If relevant, however, one approach to minimize the reconfiguration workload is to keep the number of initially configured cycles in place to the extent possible (controlled by $\beta \in [0,1]$):

$$\beta \cdot n^x \leq n_i^x \ , \ \forall x \in \boldsymbol{X} \ , \ \forall i \in \boldsymbol{S} : \ \pi_i^x = 0 \qquad (15)$$

While we express the problem as one of minimizing the number of reconfigured *p*-cycle units, the concept can easily be enhanced to express the reconfigurations in terms of reconfigured nodes or reconfigured *p*-cycles (having one or more units).

III. Case Studies

We now present results of a study network (with the COST239 topology as studied in [2]) with 11 nodes, 26 spans, and 3531 possible cycles (the average nodal degree d is 4.7). COST239 is three-connected therefore it is topologically feasible for double failures to be survived if adequate spare capacity is present.

A. *Static p-Cycles*

Figure 4 shows how R_2 can be improved using the fault sharing approach and the susceptibility approach–both of which are based on static *p*-cycles–and shows the associated capacity cost penalty. For the fault sharing approach, factor μ is varied from 1 to 10 (a μ of 1 has no effect, and a μ higher than 10 means that the protection of any span has to be shared by more than ten cycles, which seems to be impractical). For the susceptibility restriction approach, $\sigma_{\max}$ is varied from 4 to 26 (a $\sigma_{\max}$ of 3 is infeasible, and a $\sigma_{\max}$ of 26 means no effect). Figure 4 also includes the comparison value for R_2 reconfigurable networks.

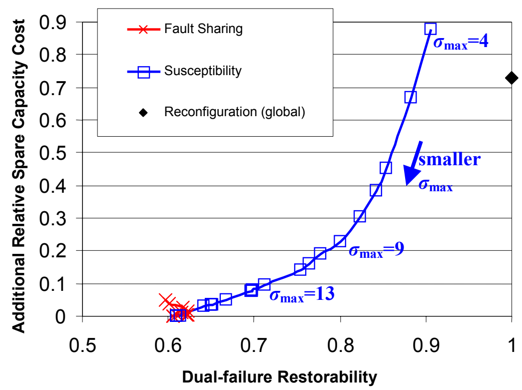

Figure 4. Cost over dual-failure restorability R_2 for three *p*-cycle design approaches.

The capacity optimal design without R_2 optimization results in a reference cost C_{ref} of 74570 and an R_2 of 61 percent. As we can see, the fault sharing approach does not bring any significant improvement of R_2. As it turns out, the fault sharing approach can only improve worst cases, but not the average case. The susceptibility restriction, however, is a tool to improve R_2: with every decrement in the maximum number of allowed protection relationships per cycle we get a significant gain in restorability. Similar to the results of [6], minimizing the maximum susceptibility such that the cost C_{ref} are preserved, however, yields only in very small restorability deviations (about 1% only). Both for the span sharing factor and the maximum susceptibility, the higher the value, the more cycles will be selected for the optimal solution. But for all the corresponding instances of Figure 4, the number of cycles is at most 15 only. We get similar results for three other test networks.

B. *Reconfigurable p-Cycle Designs for 100% Dual-Failure Restorability*

Figure 5 shows a comparison of the cost of the designs for static, non-R_2-optimized *p*-cycles (1st bar) and reconfigurable *p*-cycles (bars 2 to 7). All designs which ensure R_2 = 100% (bars 3 to 5, and bar 7) require at least 70% more capacity than the reference R_1 design. An intermediate value is reached for the case where *only the working capacity* is protected against dual failures (see Section II.C), with 60% additional cost (2nd bar). Only a marginal cost increase for the global reconfiguration design is involved, if at most 5% of the *p*-cycles can change (4th bar). Compared with the global reconfiguration output (3rd bar), 47% more cost is needed for keeping *p*-cycles untouched and adding new ones (7th bar). If existing *p*-cycles are exploited, the cost increase is only 2% (5th bar). With a cost increase of 27%, again an intermediate value is reached for the case where only the working capacity is protected against dual failures and existing *p*-cycles are untouched (6th bar).

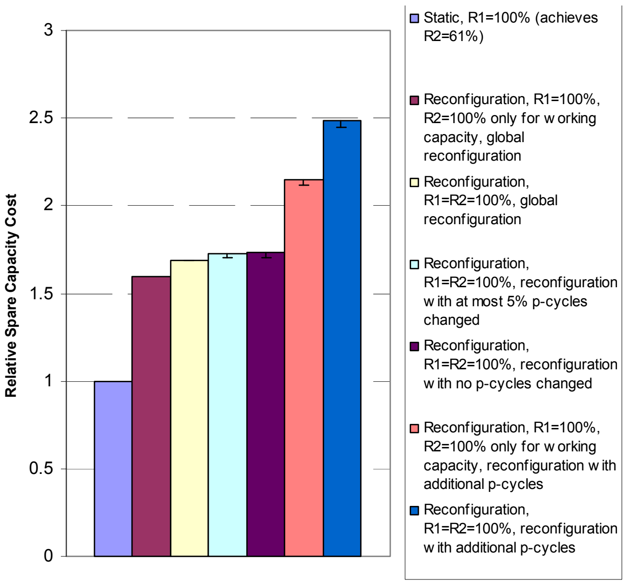

Figure 5. Cost comparison between designs for static and reconfigurable *p*-cycles (for bars 4 to 7 only approximate solutions, the intervals indicate the LP relaxation bounds).

Although a capacity cost increase of 70% for an R_2 design relative to the R_1 design sounds high, this value may be expected. In addition, given the low initial redundancy of the R_1 reference *p*-cycle design, this still represents R_2 restorable networks that typically will have less capacity than ring or 1+1 APS-protected R_1 networks. From the capacity point-of-view, an efficient R_1 design for COST239 could achieve a spare to working capacity ratio of $1/(d\text{-}1) = 27\%$ (see for example [12] for the derivation of this formula). For an efficient R_2 design, however, the estimation climbs to $2/(d\text{-}2) = 74\%$ (during a double span failure incident at a node, the working capacity of the 2 spans is best spread evenly on the node's surviving d-2 spans). Therefore, if we relate these capacity estimations for the two designs, we expect approximately 2.7 times more spare capacity needed for $R_2 = 1$ compared with $R_1 = 1$. For the test network, however, as we deal with cost weighted spare capacity in (1) and we do not deal with an estimator value for a pure-optimal R_1 design, the cost difference between $R_2 = 1$ and $R_1 = 1$ can be lower.

IV. Outlook

In this paper, we presented capacity design methods for improved or complete dual-failure restorability using either static *p*-cycles or *p*-cycles that can be reorganized after the event of a first failure. Test case studies show that using reconfigurable *p*-cycles allows one to offer a dual-failure restorability guarantee at a significantly lower capacity cost, although this cost is still 70% higher than the cost of design for single failure restorability only.

Another approach that could be of interest for a *p*-cycle network that is optimally designed to protect against single failures, would be to consider how much the ability to reconfigure *p*-cycles can enhance the dual-failure restorability (without adding any new capacity.) Although this approach may not offer a high gain for the static case [6], reconfigurable networks may take advantage of the common pool of protection capacity to achieve higher average or minimal dual-failure restorability. The approach can be used in complete *p*-cycle reconfiguration and for static *p*-cycles with new protection relationships in the reconfiguration and is for further study. Future work on the present topic could also investigate the possibility of designing a *p*-cycle network for multiple protection classes (R_1 and R_2) as presented in [13] in the context of span-restorable networks.

The authors thank M. C. Scheffel for his helpful comments on the paper.

References

[1] W. D. Grover and D. Stamatelakis, "Bridging the ring-mesh dichotomy with *p*-cycles," in *Proc. of DRCN 2000*, pp. 92-104.

[2] D. A. Schupke, C. Gruber, and A. Autenrieth, "Optimal configuration of *p*-cycles in WDM networks," in *Proc. of ICC 2002*, pp. 2761-2765.

[3] W. D. Grover, "*p*-Cycles," Chapter 10 in *Mesh-based Survivable Networks: Options for Optical, MPLS, SONET and ATM Networking*, Prentice-Hall, Aug. 2003.

[4] M. Clouqueur and W. D. Grover, "Availability analysis of span restorable mesh networks," *IEEE JSAC*, May 2002, pp. 810-821.

[5] D. A. Schupke, A. Autenrieth, and T. Fischer, "Survivability of multiple fiber duct failures," in *Proc. of DRCN 2001*, pp. 213-219.

[6] D. A. Schupke, "The tradeoff between the number of deployed *p*-cycles and the survivability to dual fiber duct failures," in *Proc. of ICC 2003*, pp. 1428-1432.

[7] D. A. Schupke, "Multiple failure survivability in WDM networks with *p*-cycles," in *Proc. of ISCAS 2003*, vol. 3, pp. III/866-869.

[8] C. G. Gruber and D. A. Schupke, "Capacity-efficient planning of resilient networks with *p*-cycles," in *Proc. of Networks 2002*, pp. 389-396.

[9] W. D. Grover and D. Stamatelakis, "Cycle-oriented distributed preconfiguration: ring-like speed with mesh-like capacity for self-planning network restoration," in *Proc. ICC 1998*, pp. 537-543.

[10] D. A. Schupke, M. Jaeger, and R. Huelsermann, "Comparison of resilience mechanisms for dynamic services in intelligent optical Networks," in *Proc. of DRCN 2003*, pp. 106-113.

[11] H. Hwang, S. Ahn, Y. Yoo, and C. S. Kim, "Multiple shared backup cycles for survivable optical mesh networks," in *Proc. of ICCCN 2001*, pp. 284-289.

[12] J. Doucette, W. D. Grover, "Comparison of mesh protection and restoration schemes and the dependency on graph connectivity," in *Proc. of DRCN 2001*, pp. 121-128.

[13] M. Clouqueur, W. D. Grover, "Mesh-restorable networks with complete dual failure restorability and with selectively enhanced dual-failure restorability properties," in *Proc. of OptiComm 2002*, pp. 1-12.