Abstract — Dual-failure scenarios are a real possibility in today’s optical networks and it is becoming more and more important for carriers and network operators to consider them when designing their networks. In this paper, we develop and analyze a linear programming model to design a p-cycle network to meet a user-specified minimum dual failure restorability. Results show that p-cycle networks designed for single-failure restorability only exhibit some inherent dual-failure restorability, and that explicitly providing dual-failure restorability is less costly in more richly connected networks.

Index Terms — network design and optimization, network restoration and protection, optical networks

I. INTRODUCTION

A modern optical transport network is required to include an assurance of nearly immediate 100% restoration (or protection) of all working traffic affected by a cable cut, optical amplifier failure, etc. Designing for 100% restorability generally means that all of the failed working channels (traffic-bearing lightwave links forming parts of end-to-end lightpaths) or entire working lightpaths can be restored by replacement paths either end-to-end across the network or through detour-like path segments formed between the end-nodes of the failed span itself. The design process must also include explicit placement of spare capacity distributed throughout the network, and doing so at optimal cost is a frequent design goal.

The simplest and most basic form of network survivability scheme possible is diversely routed 1+1 Automatic Protection Switching (1+1 APS) [1]. 1+1 APS is an end-to-end path protection mechanism that uses head-end bridging to switch from a primary working channel to a dedicated physically diverse backup channel upon failure of any link on the primary. Due to the non-sharing nature of 1+1 APS protection, it requires a commitment of at least 100% capacity redundancy. Despite this, 1+1 APS is suitable for use with simple point-to-point terminals for either single-hop or multi-hop channels, or where simple or very fast protection protocols are needed.

Survivable rings are also quite simple, being cyclic structures formed by pre-connecting transmission systems into a closed-loop arrangement connected by inexpensive add-drop multiplexing (ADM) devices. Rings can provide two diverse routes between any two nodes on the ring, so any failure is protected by rerouting in the other direction on the ring [2]. Rings gained in prominence during the 1980s, and are now widely used, particularly in metro area networks where costs are usually dominated by nodal devices. However, their capacity redundancy is quite high (minimum of 100%), so other methods have been developed that prove more promising in scenarios where costs are dominated by fibre and related rights-of-way (regional and long haul networks, for instance).

Shared mesh survivability makes use of more expensive optical cross-connect (OXC) nodal devices, but exhibits much better capacity redundancies (50-150% is typical) [4]. Span restoration is rather simple mesh survivability mechanism, where backup routes are formed between the immediate end-nodes of a failed span [3]. Relative to end-to-end path-oriented mechanisms (discussed next), restoration routes in span restoration tend to be shorter and more localized around the failure, so such mechanisms typically provide fast recovery speeds and exhibit easier control of optical path transmission effects (power levels, noise impairments, dispersion, etc.).

With the continuing evolution of optical networking and the prominence of IP networks, end-to-end shared backup path protection (SBPP), also known as failure independent shared path protection, has emerged as a capacity-efficient alternative to 1+1 APS [5]-[6]. This is particularly true in IP networks through the use of MPLS protocols. Dynamic path restoration (or failure-dependent path protection, as it is often known) is similar except that any given working lightpath might make use of any number of end-to-end backup lightpaths, with the choice of backups being dependent on the location of the failure [7].

p-Cycles are a more recent form of mesh network survivability mechanism, combining the speed of ring networks with the capacity efficiency of mesh networks [8]-[10]. Like rings, a p-cycle protects working channels by providing loop-back protection paths over a pre-connected cyclic structure. A p-cycle differs from rings in that it can also protect working channels on spans not on the cycle itself. More specifically, so-called straddling spans, which are not on the cycle but whose end-nodes are, are offered a protection path in each direction around the p-cycle. p-Cycle restoration that will be the main focus of this paper.

II. BACKGROUND AND MOTIVATION

As described above, p-cycles are ring-like pre-configured (i.e., pre-connected) structures of spare capacity, and are capable of protecting on-cycle and straddling span failures. Upon failure of a protected span, the restoration mechanism “breaks into” the p-cycle and working channels on the failed...
span are rerouted around the failure. \( p \)-Cycles differ from rings in one fundamental way. The key difference, and the main source of \( p \)-cycles’ increased efficiency, is the protection of the straddling spans. While a unit-sized ring can only protect a single working channel on each span on the ring itself, a \( p \)-cycle is also able to protect two working channels on each straddling span (it’s two because one channel can be restored in each direction around the \( p \)-cycle). Because of their ability to protect straddling spans, \( p \)-cycles are significantly more efficient than rings, yet they share the same rapid restoration as rings owing to their pre-connection prior to failure. A \( p \)-cycle and sample restoration processes are illustrated in Figure 1.

![Figure 1 - An illustration of \( p \)-cycle restoration.](image)

Another key difference between rings and \( p \)-cycles is that the latter are spare-capacity-only structures. This allows working lightpaths to be shortest path routed or otherwise routed as desired through the network graph. Working paths on a straddling spans do not even need to be routed within any \( p \)-cycle at all to be protected as long as each span crossed is a straddler, and if the network is designed appropriately, some spans don’t require any spare capacity at all (i.e., they are 100\% revenue earning working capacity spans). In a ring network, on the other hand, working lightpaths are constrained by the systems used to protect them, and their routes must be carefully matched to the ring structures and inter-ring transitions.

Prior work on \( p \)-cycle networks has focused on numerous topics, including wavelength continuity and conversion [10], IP layer \( p \)-cycles [11], reliability [12], and algorithmic design [13]. \( p \)-Cycles have also been adapted for end-to-end path protection [14]-[15], node-failure protection [16], multi-service optical networks [17], and multi-domain networks [18]. There has been very little work done on dual-failure (or generally multi-failure) \( p \)-cycle networks, however. Work in [19] studied \( p \)-cycles and shared-risk link groups (SRLGs) where a \( p \)-cycle network is designed to protect spans that share a common cause of failure, but the SRLGs were known and specifiable in advance. In [20], multiple node failures were explicitly addressed, but the design model didn’t necessarily reach any particular levels of dual-failure restorability. Dual failures were also addressed in [21], which explicitly designed for dual failure restorability of specified “gold” class working lightpaths. But the vast majority of the existing literature focuses on efficiently designing \( p \)-cycle networks for a guarantee of 100\% restoration of any single span failures. We generally denote such a network as having \( R_1 = 1 \) where \( R_1 \) is its single failure restorability. While this target may be the most efficient and cost-effective approach to providing survivability to a network, it makes no assurance of any particular dual-failure restorability, which we denote as \( R_2 \). Our goal in the present work is to extend existing design methods to the design of a \( p \)-cycle network with an assured survivability to any single failure (i.e., \( R_1 = 1 \)) while also assuring some minimum specified level of \( R_2 \).

### A. Why the Concern for Dual Failures?

Basic probability theory will tell us that by far the most common class of span failure states in a typical optical network is just a single span failure in isolation, not overlapping in time with any other failure. In those events, any \( R_1 = 1 \) network will remain fully operational. It is dual-failure events, however, that will make the largest contribution to network outage and unavailability [22], and such occurrences arise frequently enough to be of concern to most network operators, particularly those in need of high-availability services. In networks with route-mileages in the tens of thousands of km, and the current per-km failure rates [23], basic statistical considerations will show that several dual-failure events will arise each year. Introducing maintenance and upgrading to the mix only serves to complicate the matter, since any action that takes a fibre span out of service can be thought of as equivalent to a failure for these purposes [24].

### III. Dual-Failure Restorability (\( R_2 \))

Like the work of [22], here we define the network-wide dual-failure restorability, \( R_2 \), as an average of individual \( R_2(i,j) \) values where \( R_2(i,j) \) is the restorability of working channels on spans \( i \) and \( j \) when those two spans have simultaneously failed (or when their failures have overlapped in time). We differ from that prior work, however, in how we define \( R_2(i,j) \). If \( w_i \) and \( w_j \) are the number of working channels on spans \( i \) and \( j \), respectively (and at least one of \( w_i \) and \( w_j \) is non-zero), and \( N_{ij} \) is the total number of working channels on both spans that are restorable when those spans are failed, then \( R_2(i,j) \) is the fraction those spans’ working channels that are restorable:

\[
R_2(i,j) = \frac{N_{ij}}{w_i + w_j}
\]

As stated above, we define the network-wide \( R_2 \) is equivalent to the average of the \( R_2(i,j) \) values for all pairs of spans (in the set of all spans in the network, \( S \)):

\[
R_2 = \frac{\sum_{\forall i \in S} \sum_{\forall j \in S \neq i} R_2(i,j)}{|S| \times (|S| - 1)}
\]

The main problem then becomes one of properly determining the individual \( R_2(i,j) \) values for a \( p \)-cycle network. While this may at first seem a relatively straightforward task, it is actually quite difficult (and overwhelmingly so to do analytically). Each \( R_2(i,j) \) value will depend on the detailed (and complicated) interaction of numerous factors, including the amount of working channels on each failed span, the number of \( p \)-cycles available to protect the failed spans, the precise assignment of failed working channels to specific
In this final failure combination, both failed spans, \( i \) and \( j \), are on \( p \)-cycle \( p \). Each failure, therefore, cuts the restoration route that would be used by the other, and so there is a restoration route available for either:

\[
N^3_{i,j} = 0
\]  

4) \( x_{i,p} = 1 \) and \( x_{j,p} = 2 \)

In this combination of failures, the first failed span, \( i \), is on \( p \)-cycle \( p \), while the second failed span, \( j \), is a straddling span. In this situation, the number of restoration routes available will be the minimum of \( w_i + w_j \) and \( n_p \) (using similar reasoning as that used in scenario 1). So \( N_{i,j} \) is:

\[
N^4_{i,j} = \min \left( w_i + w_j, \sum_{\forall p \in P, x_{i,p} = 1, x_{j,p} = 2} n_p \right)
\]  

5) \( x_{i,p} = 2 \) and \( x_{j,p} = 1 \)

This dual-failure combination is identical to the previous except that it is span \( j \) (the second failure) that is on the cycle and span \( i \) (the first failure) is the straddler. Therefore \( N_{i,j} \) is:

\[
N^5_{i,j} = \min \left( w_i + w_j, \sum_{\forall p \in P, x_{i,p} = 2, x_{j,p} = 1} n_p \right)
\]  

6) \( x_{i,p} = 2 \) and \( x_{j,p} = 2 \)

In this final failure combination, both failed spans, \( i \) and \( j \), are straddling spans. The total number of restoration routes used on \( p \)-cycle \( p \) to restore the failed spans in this situation is \( 2 \cdot n_p \), if \( w_i + w_j < 2 \cdot n_p \), and \( w_i + w_j \) otherwise. So \( N_{i,j} \) is:

\[
N^6_{i,j} = \min \left( w_i + w_j, \sum_{\forall p \in P, x_{i,p} = x_{j,p} = 2} 2 \cdot n_p \right)
\]  

The above scenarios are defined from the point of view of a single \( p \)-cycle, \( p \). If we want to determine the restorability with respect to a pair of span failures over all cycles that can protect one or both of those spans, we need to consider all such eligible cycles simultaneously. In fact, we can easily calculate \( R_s(i,j) \) as the minimum of the sum of the working capacities of the two failed spans or the accumulation of \( N_{i,j} \) values of all those individual scenarios, divided by the sum of the working capacities of the two failed spans:

\[
R_s(i,j) = \frac{\text{minimum of the sum of the working capacities of the two failed spans}}{\text{accumulation of } N_{i,j} \text{ values of all those individual scenarios}}
\]
\[ R_2(i, j) = \min \left( \frac{w_i + w_j \cdot N_{i,j}^1 + N_{i,j}^2 + N_{i,j}^3 + N_{i,j}^4 + N_{i,j}^5 + N_{i,j}^6}{w_i + w_j} \right) \] (9)

IV. ILP DESIGN MODELS

Using the above upper bound equation, we can then develop an ILP model for designing a minimum-cost p-cycle network with a specified minimum dual-failure restorability. To do so, we first define some key notation. \( \delta_{i,p} \in (0,1) \) is an input parameter, where \( \delta_{i,p} = 1 \) if span \( i \) is on cycle \( p \), and \( \delta_{i,p} = 0 \) if it isn’t (i.e., \( \delta_{i,p} = 1 \) if \( x_{i,p} = 1 \) and \( \delta_{i,p} = 0 \) if \( x_{i,p} \neq 1 \)). \( c_i \) is the per-unit cost of spare capacity on span \( i \). \( w_i \) and \( x_{i,p} \) are input parameters as defined above. \( R_2 \) is the minimum required network-average dual failure restorability. \( S \) is the set of all spans in the network, and the set \( P \) is as defined above (the set of eligible cycles in the network). \( s_j \) is a non-negative decision variable representing the amount of spare capacity required on span \( j \). \( n_p \), \( R_2 \), and \( N_{i,j}^x \) \( x = 1 \ldots 6 \) are non-negative decision variables, also defined above. The ILP design model is then:

Minimize:

\[ \sum_{i \in S} c_i \cdot (s_i + w_i) \] (10)

Subject to:

\[ s_j = \sum_{p \in P} \delta_{j,p} \cdot n_p \quad \forall j \in S \] (11)

\[ w_i \leq \sum_{p \in P} x_{i,p} \cdot n_p \quad \forall i \in S \] (12)

\[ R_2(i, j) \leq \left( \frac{N_{i,j}^1 + N_{i,j}^2 + N_{i,j}^3 + N_{i,j}^4 + N_{i,j}^5 + N_{i,j}^6}{w_i + w_j} \right) \quad \forall i \in S \] (13)

\[ \forall j \in S \quad \forall i \neq j \]

\[ R_2(i, j) \leq 1 \quad \forall (i, j) \in S^2 \quad \forall i \neq j \] (14)

\[ R_2 = \frac{\sum_{(i,j) \in S^2} R_2(i, j)}{|S^2| - |S|^2} \] (15)

\[ R_2 \geq R_2^* \] (16)

The objective we seek to minimize in (10) is the total cost of all working and spare capacity channels assigned to spans in the network. Note that since \( w_i \) are input parameters, the term in the summation could be reduced to \( c_i \cdot s_i \), but the objective as defined in (10) is valid nonetheless. The constraints in (11) assign sufficient spare channels on each span to accommodate the \( p \)-cycles placed on them, and the constraints in (12) ensures that all working channels are fully protected against all single span failures. The objective function and these two sets of constraints define the conventional minimum-cost single-failure \( p \)-cycle network design model. The remaining constraints are all specific to the new design model developed in the present work.

The constraints in (13) calculate an upper bound on \( R_2(i, j) \) variables, while those in (14) place an upper limit of 1 on all \( R_2(i, j) \) variables. Note that these two constraint sets act together to closely model the calculation given in equation (9), in the sense that \( R_2(i, j) \) will be allowed to be only as large as the larger of \( w_i + w_j = 1 \) and \( \frac{N_{i,j}^1 + N_{i,j}^2 + N_{i,j}^3 + N_{i,j}^4 + N_{i,j}^5 + N_{i,j}^6}{w_i + w_j} \). Since the equation in (15) calculates \( R_2 \) as the average of all \( R_2(i, j) \) and the constraint in (16) forces \( R_2 \) to be no less than a specified minimum, then each of the individual \( R_2(i, j) \) variables will be as large as they’re allowed to be in order to meet that minimum \( R_2 \).

In addition to the constraints above, the complete ILP model also includes 6 more constraints, each corresponding to \( N_{i,j}^x \) equations (3) through (8), which we will not repeat here because of page-count restrictions. It is clear that (except for (5)), these equations are non-linear, but it is easy to “linearize” them. Because the overall model itself will seek to make the \( N_{i,j} \) variables as large as they can be if needed to meet the minimum \( R_2 \), then we can replace each \( N_{i,j}^x = \min (a, b) \) equation with two constraints: \( N_{i,j}^x \leq a \) and \( N_{i,j}^x \leq b \). This won’t explicitly assign accurate values for the \( N_{i,j} \) variables, and by extension the \( R_2(i, j) \) variables, but it will properly ensure that they take values large enough to satisfy the \( R_2^* \) requirement if the network will allow it.

V. EXPERIMENTAL METHOD

A total of 35 test case networks are used. They are divided into two groups or families, with each family defined by a master network with an average nodal degree of \( \bar{d} = 4.0 \). Each successive member of a family is obtained by applying a succession of pseudo-random span removals to create a series of networks of ever-decreasing \( \bar{d} \), all with the same set of nodes. Therefore, each network is identical to the next higher-degree network of the same family except that one span has been removed. The master networks are shown in Figure 2.

The lightpath demand matrix for each master network is a full mesh with each node pair exchanging a uniform random integer number of lightpaths from 1 to 10. Each network in a family uses the same demand matrix used for the family’s master network. Working capacities are consistent with a shortest path routing of all lightpath demands. The set of eligible cycles is generated for each network by depth-first search (a simple and straightforward method that is time-efficient when finding a limited cycle set), and includes the 1000 shortest cycles that exist in the network. We use a follow-up algorithm to check each span to ensure existence of at least one eligible cycle capable of protecting it. If one doesn’t exist, another depth-first search algorithm constructs the single shortest cycles containing that span and adds it to the eligible
cycle set. However, in all of our test cases, the original 1000 cycles were sufficient to protect all spans.

All design models were implemented in the AMPL 9.0 mathematical programming language and solved with the CPLEX 9.0 MIP Solver on a 4-processor SUN UltraSparc III running at 900 MHz with 16 GB of RAM. All \( s_i, n_p \) and \( N_{ij} \) decision variables are integer. Results are based on full CPLEX terminations with “MIPGAP” settings of 0.0025, meaning solutions are guaranteed to be within 0.25% of optimal. Pre-processing took no more than 1 second for each test case and most CPLEX optimization took just seconds or a few minutes to solve. Several cases (the most richly connected networks with very high \( R_2 \) requirements), however, did take up to several minutes to solve.

![Figure 2 - Network topologies of the 15-node master network (at the left) and 20-node master network (at the right).](image)

**VI. RESULTS AND DISCUSSION**

Figure 3 and Figure 4 present the normalized total (working plus spare) capacity costs of the optimally designed members of the 15-node and 20-node network families respectively. Each curve corresponds to the member of the family of the indicated average nodal degree, and each data point in a curve represents a solved instance of the ILP problem with the minimum specified dual-failure restorability, \( 0 \leq R_2^* \leq 1 \), as shown on the horizontal axis. Although we have data for 15 members of the 15-node family (with \( 2.13 \leq \bar{d} \leq 4.0 \)) and 20 members of the 20-node family (with \( 2.1 \leq \bar{d} \leq 4.0 \)), we only show curves for 5 members of each family so that the figures appear less crowded and are easier to examine. The curves shown, however, are representative of the curves not shown, and show similar relationships.

The leftmost data point of each curve (\( R_2^* = 0 \)) is equivalent to a single-failure restorable \( p \)-cycle network (i.e., the conventional ILP design model mentioned earlier, which consists of the objective function in (10) and only the first two constraints, (11) and (12). As expected, the design costs of a network will increase with increasing \( R_2^* \), but it is also clear that there are ranges of \( R_2^* \) for which the design cost of a network will be the same as a \( p \)-cycle network requiring only single-failure restorability. More specifically, for instance, the 20-node network with \( \bar{d} = 3.60 \) costs the same to design with \( 0 \leq R_2^* \leq 0.5 \). The interpretation is that a network design for only single-failure restorability also has some inherent dual-failure restorability (at least 50% in the \( \bar{d} = 3.60 \) 20-node network).

![Figure 3 - Normalized capacity costs versus dual-failure restorability for the 15-node network family](image)

![Figure 4 - Normalized capacity costs versus dual-failure restorability for the 20-node network family](image)

Another interesting observation we can make is that the greater the connectivity of the network (i.e., larger \( \bar{d} \)), the greater that inherent level of dual-failure restorability. This is because in a more richly connected network, it is more likely that any pair of spans will be protected by separate \( p \)-cycles. In terms of the dual-failure scenarios described in Section III, there are fewer scenarios with \( x_{ijp} = 1 \) and \( x_{ijp} = 1 \) and more with at least one of \( x_{ijp} = 0 \) or \( x_{ijp} = 0 \).

At the other extreme end of the range of \( R_2^* \) values, most problems (particularly those for sparse networks) don’t have feasible solutions. While this is not unexpected, it is evidence of the fact that high levels of dual-failure restorability require greater network connectivity to provide a greater diversity of restoration options for any pair of failed spans. In very sparse networks, it becomes impossible to provide the same \( R_2 \) levels as we can in richly connected networks. Another more fundamental issue is that sparse networks also have a greater number of degree-2 nodes, whose adjacent spans are inherently un-restorable by \( p \)-cycles if they fail as a pair.
Figure 5 and Figure 6 show the same general data as in Figure 3 and Figure 4, except that here, the data points are arranged into curves for specified minimum dual-failure restorability levels, $R_2^*$, and plotted against average nodal degree on the horizontal axis. Again, to make the figures easier to examine, we only show several selected curves, which are representative of the curves not shown. While it may seem unnecessary to arrange the data in this way when we’ve already seen it and discussed it in the form above, it allows us to observe how the $R_2^*$ requirement affects the standard capacity versus connectivity curves that are becoming more common in the literature.

Figure 5 - Normalized total capacity costs versus average nodal degree for the 15-node network family

Figure 6 - Normalized total capacity costs versus average nodal degree for the 20-node network family

We can see here that the higher the $R_2^*$ requirement, the steeper and more divergent the capacity versus connectivity curves become at the lower $\bar{\delta}$ values. This observation corroborates the observation made in the analysis of the previous figures: if we wish to provide especially high levels of $R_2$, then it will become more costly faster than if we wished to provide lower levels of $R_2$. On the other hand, when the networks are richly connected (i.e., high $\bar{\delta}$ values), then all of the individual $R_2$ curves are much more closely aligned. Quantitatively, we see that in the 15-node network family, the capacity requirements for $R_2^* = 0.7$ are approximately 30% greater in the $\bar{\delta} = 4.0$ network than the capacity requirement for $R_2^* = 0$ (i.e., single-failure restorability only). In the $\bar{\delta} = 2.53$ network, capacity requirements for $R_2^* = 0.7$ are over 300% greater than for $R_2^* = 0$.

The second observation we can make is that the curves with high $R_2^*$ requirements are truncated more quickly at low connectivities. Again, this supports the observations we made in the previous two figures, that if we wish to provide high levels of $R_2$, then we can’t do so in particularly sparse networks.

We can also point out that in a few cases, one of the curves increases as we go to a higher $\bar{\delta}$ (e.g., the $R_2^* = 0.8$ curve at $\bar{\delta} = 3.4$ and $\bar{\delta} = 3.5$ in Figure 6). While this seems to go against our understanding of the behaviour of capacity versus connectivity curves, the reasons are quite simple. First, we note again that working capacities are via shortest path routing of lightpath demands. That means than the working capacities in any two members of a network family will be different, and even a very slight change in working capacities can have unpredictable effects on the resultant spare capacity requirements. Secondly, we can recall that we enumerated only the 1000 shortest eligible cycles for each test network. While we would expect that spare capacity requirements would decrease when we add a new span to the network, there is no guarantee that the new set of 1000 eligible cycles we enumerate will be as well suited to the particular network.

VII. CONCLUDING REMARKS

We have developed a new ILP model to design a fully single-failure restorable $p$-cycle network with a specified minimum dual-failure restorability as well. Results have shown that a $p$-cycle network designed for full single-failure restorability has some inherent dual-failure restorability. Even sparse networks of $\bar{\delta} = 2.4$ can have 30% or more inherent dual-failure restorability at no additional capacity cost. The more richly connected test networks have inherent dual-failure restorability in excess of 50%. In addition, capacity costs increase significantly faster with increases in $R_2^*$ in particularly sparse networks than in richly connected networks.

The design model developed here would assist carrier in offering some assured network-wide dual-failure restorability to its customers, and could even help in negotiating SLAs. Furthermore, the ability to more directly evaluate capacity and dual-failure restorability tradeoffs would provide network operators with a better fundamental understanding of their networks and the implications of dealing (or not dealing) with potential dual-failure scenarios.
REFERENCES


