Optical Switching and Networking 44 (2022) 100653

Contents lists available at ScienceDirect

Optical Switching and Networking

journal homepage: www.elsevier.com/locate/osn

Circuits/cutsets duality and theoretical foundation of a structural approach to survivable logical topology mapping in IP-over-WDM optical networks

Krishnaiyan Thulasiraman [a,1,*], Tachun Lin [b], Muhammad Javed [c], Guoliang Xue [d,1], Zhili Zhou [e]

[a] School of Computer Science, University of Oklahoma, Norman, OK, USA
[b] Computer Science & Information Systems, Bradley University, Peoria, IL, USA
[c] Department of Computing and Technology, Cameron University, Lawton, OK, USA
[d] Ira A. Fulton Schools of Engineering, Arizona State University, Tempe, AZ, USA
[e] Grubhub Inc., Chicago, IL, USA

ARTICLE INFO

Keywords:
IP-ove-WDM optical networks
Survivable logical mapping
Circuits/cutsets duality
Survivability against multiple physical link failures
Network algorithms

ABSTRACT

The survivable logical topology mapping (SLTM) problem in IP-over-WDM networks is to map each link in the logical topology (IP layer) onto a lightpath in the physical topology (optical layer) such that a failure of a physical link does not cause the logical topology to become disconnected. This problem is known to be NP-complete. For this SLTM problem, two lines of investigations have been reported in the literature: the mathematical programming approach [1] and the structural approach introduced by Kurant and Thiran in [2] and pursued by Thulasiraman et al. [3,4,5]. In this paper we present an integrated treatment of the theoretical foundation of the survivable topology mapping problem presented in [3,4,5]. We believe that the algorithmic strategy developed in this paper will serve as an important phase in any strategy in the emerging area of resilient slicing of elastic optical networks. We conclude with a comparative evaluation, based on simulations, of the different algorithmic strategies developed in the paper, and also pointing to applications beyond IP-over-WDM optical networks, in particular, survivable design of inter-dependent multi-layer cyber physical systems such as smart power grids.

1. Introduction

Wavelength-Division Multiplexing (WDM) technology is widely applied in long-haul networks because of its high bandwidth and reliability. The communication between two end nodes on a WDM network is carried out through a path, namely a *lightpath*, which utilizes a single wavelength through optical nodes like optical cross-connects and optical add-drop multiplexers without opto-electro-optical conversion on intermediate optical nodes. Most data services nowadays, like HTTP, VoIP, FTP, etc., apply a dominating protocol called Internet Protocol (IP). For an IP-over-WDM network, the traffic on each IP link is carried through a lightpath in the WDM network. For a multi-hop data transmission in IP-over-WDM network as shown in Fig. 1, the traffic on the 1-2-4 path in the IP network is implemented through two lightpaths 1–2 and 2-3-4 in the WDM network.

Given an IP-over-WDM network with physical and logical topologies $G_P = (V_P, E_P)$ and $G_L = (V_L, E_L)$, where $V_P(V_L)$ represents physical (logical) nodes/vertices and $E_P(E_L)$ represents physical (logical) edges/links, a survivable routing in such a network is usually determined by edge-disjoint lightpath routing for logical edges. If any physical link failure does not disconnect the logical topology, this routing is called a survivable routing. For this problem two lines of investigations have been pursued in the literature: the mathematical programming based approach initiated by Modiano and Narula-Tam [1], and the structural approach initiated by Kurant and Thiran [2] and pursued further by Thulasiraman et al. [3–5]. The mathematical programming approach is not scalable for large networks, though it gives considerable insight into certain important aspects of the problem. The structural approach requires contraction and expansion of logical graphs and computing link-disjoint lightpaths between pairs of vertices in the physical topology. This approach requires finding a set of mutually disjoint paths between the nodes of a small subset of logical links, and so considerably

* Corresponding author.
E-mail address: thulasi@ou.edu (K. Thulasiraman).

[1] The work of Krishnaiyan Thulasiraman has been supported by NSF grants CNS-1115130 , ANI-0312435 and ECS 04-26831 and the work of G. Xue has been supported by NSF grants 0830739 and 0721803.

https://doi.org/10.1016/j.osn.2021.100653
Received 21 May 2021; Received in revised form 2 November 2021; Accepted 17 November 2021
Available online 2 December 2021
1573-4277/

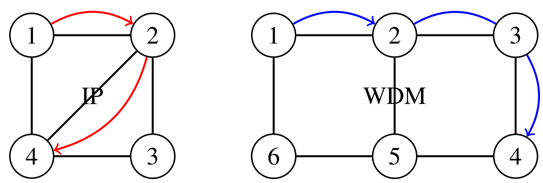


Fig. 1. An example of potential lightpaths for logical link.

reduces the complexity.

The structural approach was studied by Thulasiraman et al. [3–5] using the concept of circuit/cutset duality. In this paper we present an integrated treatment of the results in Refs. [3–5]. We next present a review of literature in this area.

2. Related work

2.1. Mathematical programming based approach

Survivability of a logical topology mapping (routing) can be guaranteed if the lightpaths in the physical topology corresponding to this mapping are all link-disjoint. Since finding disjoint paths between pairs of nodes is NP-complete [6], survivable design of the logical topology in an IP-over-WDM network is also an NP-complete problem. Modiano and Narula-Tam [1] proved necessary and sufficient conditions for survivable routing under a single failure in IP-over-WDM networks and formulated the problem as an ILP. Todimala and Ramamurthy [7] adapted the concept of SRLG and computed the routing through an ILP formulation. Extensions of the work in Ref. [1] are given in Refs. [8,9]. Lee et al. [8] introduced certain connectivity metrics for layered networks and provided ILP formulations for the lightpath routing problem satisfying these metrics. In particular, they provided approximation algorithms for lightpath routing maximizing the minimum cross layer cut metric which captures the robustness of the networks after multiple physical link failures. Kan et al. [9] discussed the relationship between survivable lightpath routing and spare capacity requirements on the logical links to satisfy the original traffic demands after failures. Lin et al. [10,11] introduced the concepts of weakly and strongly survivable routings and provided MILP formulations for generating a logical topology routing and rerouting (after a physical link failure) to maximize the total satisfied demand before and after a failure. They also considered the question of spare capacity minimization to maximize the demand satisfaction. Lin et al. [12] considered how lightpaths used for survivable routing can be combined with monitoring trails [13,14] to achieve localization of physical link failures. Zhou et al. [15] provided MILP formulations for cross-layer survivability under multiple metrics, including the cross-layer cut introduced by Lee and Modiano [8]. These formulations have been directly adopted in the study of the survivable network virtualization problem [16,17]. In Ref. [18] et al. introduced the concept of logical protecting spanning trees and provided MILP problems using column generation technique that can be adopted to study other classes of network optimization problems encountered in communication network planning and design.

2.2. Structural Approach

A common drawback of ILP approaches is that they are not scalable as the network size increases. Hence, heuristic approaches that provide approximations to the optimal solutions are presented. To handle the drawback of ILP approaches, Kurant and Thiran [2] proposed the Survivable Mapping by Ring Trimming (SMART) framework which first attempts to find link-disjoint paths for the links of a subgraph of the given logical graph. If such a mapping exists, the subgraph is contracted. The procedure is repeated until the logical graph is contracted to a single node, or at some step disjoint mappings cannot be found. In the former case, the resulting mappings define a survivable mapping of the given logical graph. Another approach proposed by Lee et al. [19] utilized the concept of ear-decomposition on bi-connected topologies. One can show that this is, in fact, a special variant of the framework given in Ref. [2], though it was developed independently. Javed et al. [20,21] obtained improved heuristics based on SMART. Using duality theory in graphs, a generalized theory of logical topology survivability was given by Thulasiraman et al. [3,4]. Thulasiraman et al. [22] considered the problem of augmenting the logical graph with additional links to guarantee the existence of a survivable mapping. It has been shown in Ref. [22] that if the physical network is 3-edge connected, survivability-guaranteed augmentation of the logical topology is always possible. An earlier work that discussed augmentation is in Ref. [23].

In [24] Deng et al. presented an MILP formulation for the SLTM problem which used a novel technique to test the connectivity of a logical topology when a physcial link fails. We believe that this formulation can serve as a good benchmark for comparing different algorithms reported in the literature. In two related works [25,26], Zhu et al. provided an MILP formulation for cloud network mapping to survivable multiple failures using the concept of protection trees introduced in Ref. [18].

Cohen and Nakibly [27] studied the problem of designing survivable topologies in the face of uncertain traffic knowledge. They also point to several works on the optial topology design problem that complement the work provided in this section.

In this paper, we present an integrated treatment of our works in Refs. [3–5] on the structural approach for the SLTM problem. In these works we generalized, using the duality between circuits and cutsets of a graph, the methodology provided by Kurant and Thiran [2] for the SLTM problem. We provided a unified algorithmic framework which included four algorithmic frameworks as special cases. We also presented results of an analytical study on the robustness of these frameworks with respect to their ability to provide logical topology mappings that survive multiple physical link failures. This review paper is self contained with adequate background material on graph theoretic concepts required to follow the developments discussed.

For graph theoretic concepts not covered in this paper see Ref. [28].

3. Circuits and cutsets duality

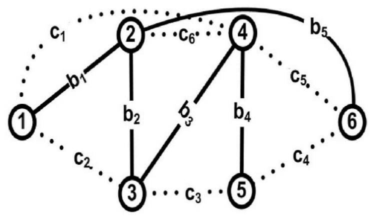
Duality between circuits and cuts in a graph has been extensively studied and plays a fundamental role in several applications [29,30]. Deleting an edge and contracting an edge are also dual operations. In this section, we present several concepts and results relating to this duality. These results provide the basis for the algorithmic frameworks presented in the following sections.

Consider a connected undirected graph $G(V, E)$ with vertex set V and edge set E. Without loss of generality, we assume that there are no parallel edges or self loops in G. Let G have $|V| = n$ vertices (or nodes) and $|E| = m$ edges (or links).

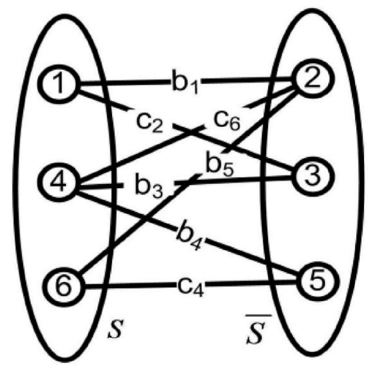
A connected acyclic subgraph of G containing all the n nodes is called a *spanning tree* T of G. The edges of a spanning tree T are called *branches* of T. The remaining edges of G are called *chords* with respect to T. We may also refer to chords as *non-tree edges*.

Consider a partition $(S, \overline{S})$ of vertex set V. Here $\overline{S}$ denotes the complement of $S \subsetneqq V$, i.e. $\overline{S} = V - S$. Then the set of edges with one node in S and the other in $\overline{S}$ is called a *cut* of G. For example, consider the graph G in Fig. 2a. Here the vertices are numbered 1, 2, …, 6. The bold edges in this figure denote the branches of a spanning tree T of G and the dotted edges are the chords of this tree. The partition $(S, \overline{S})$ with $S = \{1, 4, 6\}$ and $\overline{S} = \{2, 3, 5\}$ defines the cut shown in Fig. 2b.

Adding a chord c to a spanning tree T produces exactly one circuit. This is called the *fundamental circuit* (in short, *f-circuit*) of T with respect to the chord c. We denote this circuit as $B(c)$. For example, if we add chord c_1 to the tree in Fig. 2a we get the fundamental circuit $B(c_1)$



(a) A graph with a spanning tree (bold lines)



(b) A cut

Fig. 2. Concept of a tree and a cut.

consisting of the edges $\{c_1, b_1, b_2, b_3\}$. Similarly, if we add chord c_4 to the tree in Fig. 2a we get the fundamental circuit $B(c_4)$ that contains edges $\{c_4, b_2, b_3, b_4, b_5\}$.

Suppose we remove a branch b from a spanning tree T, then the tree T gets disconnected resulting in two trees (not spanning) T_1 and T_2. The sets of nodes in T_1 and T_2 define a partition of V. The corresponding cut is called the *fundamental cutset* (in short, *f-cutset*) of T with respect to branch b. For example, if we remove the branch b_3 from the tree T of Fig. 2a then we get trees T_1 and T_2 given by branches $\{b_1, b_2, b_5\}$ and $\{b_4\}$, respectively. The corresponding fundamental cutset $Q(b_3)$ consists of the edges $\{b_3, c_1, c_3, c_4, c_5, c_6\}$. Note that the subgraphs induced by the vertex sets of T_1 and T_2 are both connected. Cuts with this property are also called *primary cuts* [7].

Given a spanning tree T with branches $\{b_1, b_2, \ldots, b_{n-1}\}$ and chords $\{c_1, c_2, \ldots, c_{m-n+1}\}$, then the *fundamental circuit matrix* $B_f = [b_{ij}]_{(m-n+1)\times(m)}$ has one row for each chord and one column for each edge. The entry b_{ij} is defined as follows:

$$b_{ij} = \begin{cases} 1, & \textit{if } B(c_i) \textit{ contains edge } j \\ 0, & \textit{otherwise.} \end{cases}$$

Arranging the rows of B_f such that the jth row ($j \le m - n + 1$) corresponds to the fundamental circuit $B(c_j)$ and arranging the columns in the order $\{c_1, c_2, \ldots, c_{m-n+1}, b_1, b_2, \ldots, b_{n-1}\}$, we can write the B_f matrix as $B_f = [U|B_{ft}]$, where U is the unit matrix of size $(m - n + 1)$. For example, the B_f matrix with respect to the spanning tree T of Fig. 2a is given in (1).

$$\begin{array}{c|cccccc|ccccc} & c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & b_1 & b_2 & b_3 & b_4 & b_5 \\ c_1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ c_2 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ c_3 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ c_4 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ c_5 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ c_6 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \end{array} \quad (1)$$

In a similar manner the *fundamental cutset matrix* with respect to the tree T can be defined as $Q_f = [q_{ij}]_{(n-1)\times(m)}$. Q_f has $(n - 1)$ rows, one for each fundamental cutset and one column for each edge. The entry q_{ij} is defined as

$$q_{ij} = \begin{cases} 1, & \textit{if } Q(b_i) \textit{ contains edge } j \\ 0, & \textit{otherwise.} \end{cases}$$

Arranging the rows of Q_f such that the jth row corresponds to f-cutset $Q(b_j)$ and the columns correspond to edges in the order $\{b_1, b_2, \ldots, b_{n-1}, c_1, c_2, \ldots, c_{m-n+1}\}$, the Q_f matrix can be written as $Q_f = [U|Q_{fc}]$. For example, the Q_f matrix with respect to the tree T of Fig. 2a is given in (2).

$$\begin{array}{c|ccccc|cccccc} & b_1 & b_2 & b_3 & b_4 & b_5 & c_1 & c_2 & c_3 & c_4 & c_5 & c_6 \\ b_1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ b_2 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ b_3 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ b_4 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ b_5 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \end{array} \quad (2)$$

A *subgraph* (for example, a circuit or a cut) can be represented by a binary vector with m entries, one entry for each edge, and with an entry equal to 1 if the corresponding edge is present in the subgraph. Thus, rows of B_f and Q_f are the binary vectors representing the fundamental circuits and fundamental cutsets. For convenience, in the following we will use the same symbol $B(c_j)(Q(b_j))$ to denote the set of edges in a fundamental circuit (cutset) as well as the corresponding binary vector.

Proofs of the following dual results can be found in Ref. [28].

Theorem 1. *(a) If a cut contains the branches* $\{b_1, b_2, \ldots, b_j\}$ *then the corresponding cut vector can be represented as modulo 2 addition of the vectors* $Q(b_1), Q(b_2), \ldots, Q(b_j)$. *That is, the cut vector is equal to* $Q(b_1)\oplus Q(b_2)\oplus \ldots \oplus Q(b_j)$.

(b) *If a circuit contains the chords* $\{c_1, c_2, \ldots, c_j\}$ *then the corresponding circuit vector can be represented as modulo 2 addition of the vectors* $B(c_1), B(c_2), \ldots, B(c_j)$. *That is, the circuit vector is equal to* $B(c_1)\oplus B(c_2)\oplus \ldots \oplus B(c_j)$. □

Theorem 2. *(Orthogonality): A circuit and a cut have an even number of common edges.* □

Theorem 3. $B_{ft} = Q_{fc}^t$, *where* Q_{fc}^t *is the transpose of* Q_{fc}. □

The above properties can be verified using the B_f and Q_f matrices given in equations (1) and (2).

An ordered sequence $B(c_1), B(c_2), \ldots, B(c_k)$ is a *circuit cover sequence* or simply a *B-sequence* of length k if it is a maximal sequence satisfying

$$a) \left[B(c_j) - c_j - \bigcup_{p=1}^{j-1} B(c_p) \right] \ne \varnothing, 2 \le j \le k$$

$$b) \bigcup_{p=1}^{k} B(c_p) = E - \{\text{chords not in the } B - \text{sequence}\}$$

Note that for a given spanning tree and its f-circuits, there may be more than one B-sequence. For example for the fundamental circuits given in (1), the following are B-sequences:

(1) $B(c_1), B(c_3), B(c_5)$
(2) $B(c_4), B(c_1)$
(3) $B(c_6), B(c_1), B(c_4)$

Note that the order in which the $B(c_j)$'s appear matters in the definition of B-sequences. Without loss of generality we assume that $B(c_1), B(c_2), \ldots, B(c_k)$ is a B-sequence of length k. Let us define $S(c_j)$ as follows:

$$a) \ S(c_1) = B(c_1) - c_1$$

$$b) \ S(c_j) = B(c_j) - c_j - \bigcup_{p=1}^{j-1} B(c_p), \quad 2 \le j \le k$$

Then the submatrix of the f-circuits comprised of the rows corresponding to $B(c_1), B(c_2), \ldots, B(c_k)$ will have the structure shown in (3). Note here that × means 0 or 1.

An ordered sequence $Q(b_1)$, $Q(b_2)$, …, $Q(b_k)$ is a *cutset cover sequence* or simply a *Q-sequence* of length k if it is a maximal sequence satisfying

$$a)\ \left[Q(b_j) - b_j - \bigcup_{p=1}^{j-1} Q(b_p)\right] \neq \varnothing, 2 \leq j \leq k$$

$$b)\ \bigcup_{p=1}^{k} Q(b_p) = E - \{\text{branches not in the } Q-sequence\}$$

Note that for a given spanning tree and its f-cutsets, there may be more than one Q-sequence. For example, for the fundamental cutsets given in (2), the following are Q-sequences:

(1) $Q(b_4), Q(b_5), Q(b_3)$
(2) $Q(b_4), Q(b_5), Q(b_1), Q(b_2)$
(3) $Q(b_1), Q(b_2), Q(b_4)$

Without loss of generality, we assume that $Q(b_1)$, $Q(b_2)$, …, $Q(b_k)$ is a Q-sequence of length k. Let us define $\widehat{S}(b_j)$ as follows:

$$a)\ \widehat{S}(b_1) = Q(b_1) - b_1$$

$$b)\ \widehat{S}(b_j) = Q(b_j) - b_j - \bigcup_{p=1}^{j-1} Q(b_p), 2 \leq j \leq k$$

Then the submatrix of the f-cutsets comprised of the rows corresponding to $Q(b_1)$, $Q(b_2)$, …, $Q(b_k)$ has a structure similar to (3) as shown in (4).

The following dual results are a consequence of the structures in (3) and (4).

$$\begin{array}{c} \\ c_1 \\ c_2 \\ \vdots \\ c_j \\ c_k \end{array}
\begin{array}{cccccc|cccc|ccc|ccc|ccc|ccc}
c_1 & c_2 & \cdots & c_j & \cdots & c_k & & S(c_1) & & & & S(c_2) & & & \cdots & & & S(c_j) & & & S(c_k) & \\
1 & 0 & \cdots & 0 & \cdots & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & \cdots & 0 & \cdots & 0 & \times & \times & \times & \times & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & \cdots & 1 & \cdots & 0 & \times & \times & \times & \times & \times & \times & \times & \times & \times & \times & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & \cdots & 0 & \cdots & 1 & \times & \times & \times & \times & \times & \times & \times & \times & \times & \times & \times & \times & \times & 1 & 1 & 1
\end{array} \quad (3)$$

$$\begin{array}{c} \\ b_1 \\ b_2 \\ \vdots \\ b_j \\ b_k \end{array}
\begin{array}{cccccc|cccc|ccc|ccc|ccc|ccc}
b_1 & b_2 & \cdots & b_j & \cdots & b_k & & \hat{S}(b_1) & & & & \hat{S}(b_2) & & & \cdots & & & \hat{S}(b_j) & & & \hat{S}(b_k) & \\
1 & 0 & \cdots & 0 & \cdots & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & \cdots & 0 & \cdots & 0 & \times & \times & \times & \times & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & \cdots & 1 & \cdots & 0 & \times & \times & \times & \times & \times & \times & \times & \times & \times & \times & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & \cdots & 0 & \cdots & 1 & \times & \times & \times & \times & \times & \times & \times & \times & \times & \times & \times & \times & \times & 1 & 1 & 1
\end{array} \quad (4)$$

Theorem 4. *(a) Given a B-sequence* $B(c_1)$, $B(c_2)$, …, $B(c_k)$, *let* $B(c_{i_1})$, $B(c_{i_2}), \ldots, B(c_{i_\ell})$ *be a subsequence of this sequence then* $S(c_{i_\ell}) \subseteq B(c_{i_1}) \bigoplus B(c_{i_2}) \bigoplus \ldots \bigoplus B(c_{i_\ell})$.

(b) *Given a Q-sequence* $Q(b_1)$, $Q(b_2)$, …, $Q(b_k)$, *let* $Q(b_{i_1}), Q(b_{i_2}), \ldots, Q(b_{i_\ell})$ *be a subsequence of this sequence then* $\widehat{S}(b_{i_\ell}) \subseteq Q(b_{i_1}) \bigoplus Q(b_{i_2}) \bigoplus \ldots \bigoplus Q(b_{i_\ell})$. □

Deletion of an edge and *contraction* of an edge are dual operations. Here by contraction of an edge we refer to the operation of identifying the end vertices of the edge (short-circuiting the end vertices) and removing self loops that result from this short-circuiting. flushleft

Given a B-sequence, the submatrix of the B_f matrix that results after removing the rows that do not correspond to the chords in the B-sequence is called a *B-sequence matrix*. For example the B-sequence matrix corresponding to the B-sequence $B(c_1)$, $B(c_3)$, $B(c_5)$ is shown in (5).

$$\begin{array}{ccc|ccccc}
c_1 & c_3 & c_5 & b_1 & b_2 & b_3 & b_4 & b_5 \\
1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 & 1
\end{array} \quad (5)$$

It can be shown that deletion of a row from B_f matrix corresponds to the deletion of the corresponding chord from the graph.

Given a Q-sequence, the submatrix of the Q_f matrix that results after removing the rows that do not correspond to the branches in the Q-sequence is called a *Q-sequence matrix*. For example the Q-sequence matrix corresponding to the Q-sequence $Q(b_4)$, $Q(b_5)$, $Q(b_2)$, $Q(b_1)$ is shown in (6).

$$\begin{array}{cccc|cccccc}
b_4 & b_5 & b_2 & b_1 & c_1 & c_2 & c_3 & c_4 & c_5 & c_6 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0
\end{array} \quad (6)$$

It can be shown that deletion of a row from the Q_f matrix corresponds to contraction of the corresponding branch from the graph.

The following dual results are easy to verify.

Theorem 5. *a) The B-sequence matrix corresponding to a B-sequence is the fundamental circuit matrix of the graph that results after deleting the chords that do not appear in the B-sequence.*

b) *The Q-sequence matrix corresponding to a Q-sequence is the fundamental cutset matrix of the graph that results after contracting the branches that do not appear in the Q-sequence.* □

The *incidence set* of a vertex v in a graph is the set of vertices adjacent to that vertex. The incidence set of vertex v will be denoted by $INC(v)$. Each incident set is a cut of the graph. It is known that any set of $n-1$ incidence sets can be used to generate any cut in a graph. The *incident vector* $INC(v)$ of a vertex v is the binary vector of m entries with jth entry being 1, if the jth edge is in the corresponding incident set. The following result is a special case of Theorem 1(a). Note that we use $INC(v)$ to denote both the incidence set and the corresponding binary vector. *Incidence matrix* of G is the matrix of incidence vectors of G.

Theorem 6. *The cut vector corresponding to the cut* $(S, \overline{S})$ *can be obtained as the modulo 2 addition of the incident vectors of the vertices in S as well as the modulo 2 addition of the incidence vectors of the vertices in* $\overline{S}$. □

A sequence of $INC(v_1)$, $INC(v_2)$, …, $INC(v_k)$ is an *incident cover sequence* or simply an *INC-sequence* of length k if it is a maximal sequence satisfying

$$a)\ \left[INC(c_j) - \bigcup_{p=1}^{j-1} INC(v_p)\right] \neq \varnothing, 2 \leq j \leq k$$

$$b)\ \bigcup_{p=1}^{k} INC(v_p) = E$$

Given an *INC*-sequence, the submatrix of the incidence matrix consisting of the rows corresponding to the vertices in the *INC*-sequence has a structure similar to the structure in (4).

The following result will be used in the Proof of correctness of all algorithms developed in the following sections.

Theorem 7. [28] *A graph is connected if and only if every cut of the graph contains at least one edge.* □

4. CIRCUIT-SMART: the primal algorithm for survivable logical topology routing

Given a logical topology G_L and a physical topology G of an optical network, the *SMART* algorithmic framework given in Ref. [2] provides a methodology for finding survivable mappings of the edges of G_L into lightpaths in G. To begin with, let us call G_L the current graph.

Algorithm 1
Algorithm SMART

1: Search for a survivable subgraph of the current graph
2: **if** no such subgraph is found **then**
3: terminate SMART unsuccessfully
4: **else**
5: contract the edges of the subgraph
6: **end if**
7: **if** the current graph is a single vertex **then**
8: terminate SMART successfully
9: **else**
10: return to step 1 and use the contracted graph as the current graph
11: **end if**

When Algorithm 1 terminates successfully, the edges mapped by *SMART* provide a survivable subgraph of G_L. All the edges that were not mapped by *SMART* can be mapped arbitrarily without affecting the survivability of G_L. It can be shown that the subgraph chosen in step 1 above must be 2-edge connected for the correctness of the algorithm.

Since a circuit is the smallest 2-edge connected graph, usually in step 1 a circuit is selected for survivable mapping.

Now consider the graph G_L in Fig. 3. If the circuits are selected in the sequence C_1, C_2, C_3, C_4 and $\{e_6, e_{10}, e_{12}, e_{17}, e_{16}, e_{19}, e_{20}, e_{23}, e_3\}$ then these are fundamental circuits of the tree for which $\{e_1, e_8, e_{21}, e_{19}, e_{15}, e_6\}$ are chords. This observation is true for any choice of circuits selected and mapped by SMART. So our new algorithm CIRCUIT-SMART starts with a set of fundamental circuits and a B-sequence constructed from these circuits. Since our interest is to guarantee survivability we add to G_L new edges in parallel to some of the edges in G_L whenever necessary. The new edge added in parallel to edge e of G_L will be denoted as e'. Both e and e' will be mapped as disjoint lightpaths in G. These edges will be called *protection edges*.

Algorithm 2
Algorithm CIRCUIT-SMART

Input:	A 2-edge connected physical topology G, logical topology G_L,a spanning tree T of G_L, a set of fundamental circuits and B-sequence $B(c_1)$, $B(c_2)$, …, $B(c_k)$
Output:	A survivable logical graph $G_{L}^{\prime}$ containing G_L
1:	**for** $i = 1, 2, \ldots, k$ **do**
2:	Map a maximum subset of edges in $S(c_k)\bigcup c_k$ into disjoint lightpaths in G [31]
3:	To all other edges in $S(c_k)\bigcup c_k$, add protection edges and map each edge and its protection edges into disjoint lightpaths in G
4:	**end for**
5:	Map all the chords not in the B-sequence into lightpaths in G arbitrarily

Theorem 8. *The graph $G_L^{'}$ of edges mapped by algorithm CIRCUIT-SMART forms a survivable logical graph.*

Proof. *We prove that $G_L^{'}$ is survivable by showing that after failure of any edge in the physical topology each cut in $G_L^{'}$ satisfies the condition in* Theorem 7. *Consider a cut in $G_L^{'}$. If any edge in this cut is a protection edge e', then this edge and the corresponding edge e in $G_L^{'}$ are mapped by the algorithm into disjoint lightpaths. Hence, one of them will remain in the cut after a single physical edge failure, thereby satisfying the condition in* Theorem 7. *If there is no protection edge in the cut, then consider the branches in the cut. The cut must contain at least one branch of T because chords alone cannot form a cut. Note that each branch is in a unique $S(c_i)$. Let chord c_j be the chord with the smallest index in the B-sequence such that $S(c_j)$ contains one of the branches, say branch b_i, in the selected cut. If $S(c_j)$ contains two or more branches in the cut, then these branches are mapped by the algorithm into disjoint paths and so the cut will satisfy the condition of* Theorem 7 *after a single link failure. If $S(c_i)$ contains only one branch of the selected cut, say b_i, the cut must contain chord c_i because the cut and $B(c_i)$ must contain an even number of edges in common* (Theorem 2). *Since b_i and c_i are mapped by the algorithm into disjoint paths in the physical topology, one of these two edges will remain in the cut after a single edge failure, satisfying again the condition of* Theorem 7. *Thus in all cases, the cut will satisfy the condition of* Theorem 7 *and so the graph $G_L^{'}$ and the routing generated by CIRCUIT-SMART are survivable.* □

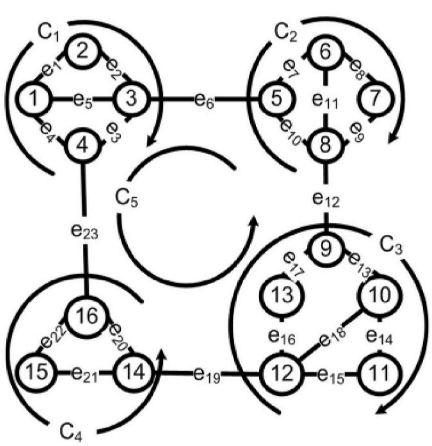


Fig. 3. Illustration of the SMART algorithm.

The essential difference between SMART and CIRCUIT-SMART is that instead of searching for a survivable circuit in each step (as in SMART), CIRCUIT-SMART uses a set of fundamental circuits. Since not all edges in a $S(c_i)$ set may be mapped in a disjoint manner, we add protection edges appropriately. The longer the B-sequence, the more are the number of chords in the survivable logical subgraph. On the other hand, a smaller B-sequence may increase the sizes of $S(c_i)$-sets and hence may result in more number of protection edges. These are considerations that must be taken into account while selecting the spanning tree.

5. CUTSET-SMART: the dual algorithm

We now present algorithm *CUTSET-SMART* (Algorithm 3) that is the dual of algorithm CIRCUIT-SMART. In the following a branch is unmatched if it is not in the given Q-sequence.

Algorithm 3
Algorithm CUTSET-SMART

Input:	A 2-edge connected physical topology G, logical topology G_L a spanning tree T of G_L, a set of fundamental cutsets and a Q-sequence $Q(b_1)$, $Q(b_2)$, …, $Q(b_k)$
Output:	A survivable logical $G_{L}^{\dprime}$ containing G_L
1:	**for** $i = 1, 2, \ldots, k$ **do**
2:	Map a maximum subset of edges in $\hat{S}(b_{k})\bigcup b_{k}$ into disjoint lightpaths in G [31]
3:	To all other edges in $\hat{S}(b_{k})\bigcup b_{k}$, add protection edges and map each edge and its protection edge into disjoint lightpaths in G
4:	**end for**
5:	To each unmatched branch b, add a protection edge b' and map them into disjoint lightpaths in G

Let G_L'' be the graph of logical edges (including protection edges) mapped by *CUTSET-SMART*.

Theorem 9. a) *The graph G_L'' and the mappings generated by CUTSET-SMART are survivable.*

b) *The graph obtained from G_L'' by contracting the branches not in the Q-sequence is survivable*

Proof. a) *Consider a cut in G_L''. If any edge in this cut is a protection edge, then this edge and the corresponding edge in G_L are mapped by the algorithm CUTSET-SMART into disjoint lightpaths. Hence one of them will remain in the cut after a single physical edge failure, thereby satisfying the condition in* Theorem 7. *If there is no protection edge in the cut, then consider the branch b_j in the cut that has the highest index in the Q-sequence. Then by* Theorem 4*(b), the set $\widehat{S}(b_j)$ will be in the cut. Since the branch b_j and the edges in the set $\widehat{S}(b_j)$ are mapped in disjoint manner by the algorithm, the cut will contain at least one edge after a physical edge failure, thereby satisfying the condition of* Theorem 7. *Thus G_L'' is survivable.*

b) *The graph obtained from G_L'' by contracting the branches not in the Q-sequence has no cut that contains the contracted tree branches and the corresponding protection edges. The result follows from the Proof of* (*a*). □

Note that Theorem 9(b) is the dual of the result that graph $G_L^{'}$ generated by *CIRCUIT-SMART* is survivable.

A closer look at the above Proof will show that in steps 1–3 of

CUTSET-SMART, it is sufficient to map in disjoint manner each branch b_i with some chord in the set $\widehat{S}(b_i)$. This results in algorithm *CUTSET-SMART-SIMPLIFIED* shown in algorithm 4. This algorithm requires finding disjoint mappings for only certain pairs of vertices in the physical topology.

Algorithm 4
Algorithm CUTSET-SMART-SIMPLIFIED

Input: A 2-edge connected physical topology G, logical topology G_L, a spanning tree T of G_L, a set of fundamental cutsets and a Q-sequence $Q(b_1)$, $Q(b_2)$, …, $Q(b_k)$
Output: A survivable logical graph $G_{L}^{\dprime}$ containing G_L
for $i = 1, 2, \ldots, k$ **do**
  Map b_i in disjoint manner with some chord in set $\hat{S}(b_{i})$ [31]
  if this is not possible for any chord in $\hat{S}(b_{i})$ **then**
    Add a protection edge for one of the chords and map the chord and its protection edge in disjoint manner
  end if
end for
To each unmatched branch b, add a protection edge b' and map them as disjoint lightpaths in G
Map all the unmapped logical edges arbitrarily

Using a result in Ref. [31] we have the following Theorem.

Theorem 10. *Given any Q-sequence of length k, algorithm CUTSET-SMART-SIMPLIFIED finds a survivable mapping of a logical topology with at most* $n - k - 1$ *protection edges, if the physical topology is 3-edge connected.* □

The main reason for the above result is that any pair of edges $\{(s_1, t_1), (s_2, t_2)\}$ in G_L can be mapped into disjoint lightpaths, if the physical topology is 3-edge connected. This means that protection edges will be needed only for the branches not in the Q-sequence.

The Proof of the Theorem 9 shows that every cut obtained from G_L'' by contracting the branches not in the Q-sequence will contain at least $\min\{|\widehat{S}(b_i)|+1, i=1,2,\ldots,k\}$ edges after a single edge failure in the physical topology. This property of *CUTSET-SMART* will be of great help in protecting the logical topology when multiple edge failures occur in the physical topology. This property is not true in the case of algorithm *CUTSET-SMART-SIMPLIFIED* though it is computationally superior to *CUTSET-SMART* and is likely to require less number of protection edges. An interesting consequence of algorithm *CUTSET-SMART-SIMPLIFIED* is the following.

Theorem 11. *The structure shown in Fig. 4 is survivable, if the physical topology is 3-edge connected.* □

This Theorem is a consequence of the following: given any three vertices x, y, and z in a 3-edge connected graph G, there exist edge-disjoint paths P_{xy}, P_{yz}, P_{xz}. This means that we can map b_{n-2}, b_{n-1}, and c_{n-2} in disjoint manner. flushleft

An interesting application of this result is as follows. Note that protection edges are used in the algorithms of Sections 4 and 5 to guarantee survivability. Consider a set of edges that form a path and require protection edges, then it can be shown that we can augment the logical topology by adding new logical edges c_1, c_2, …as in Fig. 4, instead of protection edges (that is, new parallel logical edges) and guarantee survivability. This is the topic of augmentation studied in Refs. [22,23].

Note that we have not been able to prove a property similar to the one in Theorem 10 in the case of *CIRCUIT-SMART*. Nor has it been possible for us to find a simplified version of *CIRCUIT-SMART* akin to *CUTSET-SMART-SIMPLIFIED*.

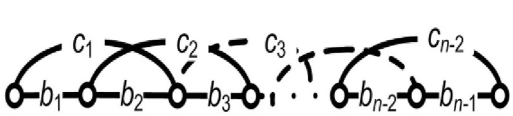


Fig. 4. A survivable network structure.

6. INCIDENCE-SMART

Incident sets are special cases of cuts. So an algorithm similar to *CUTSET-SMART* can be designed. Instead of explicitly starting with a spanning tree of G_L, we present an algorithm which reflects the unified framework in terms of *INC*-sets defined in Section 3. Note that for any *INC*-sequence there is at least one vertex that is not in the sequence. Let us call one such vertex as *datum*.

In the following algorithm (Algorithm 5) the given G_L will be the initial current graph.

Algorithm 5
Algorithm INCIDENCE-SMART

Input: A 2-edge connected physical topology G, a logical topology G_L, *INC*-sequence $INC(v_1)$, $INC(v_2)$, …, $INC(v_k)$
Output: A survivable logical graph $G_{L}^{\dprime}$ containing G_L
for $i = 1, 2, \ldots, k$ **do**
  if vertex v_i has degree greater than or equal to 2 in the current graph
    Map all the edges incident on v_i into disjoint lightpaths in G and remove v_i from the current logical graph
  end if
  if the degree of v_i in the current graph is one **then**
    Add a new logical edge connecting v_i to the datum vertex
    Map this new edge and the only edge incident on v_i into disjoint lightpaths and v_i from the logical graph
  end if
  if degree of v_i in the current graph is zero **then**
    Add two new parallel logical edges connecting v_i to the datum vertex. Then map these two edges into disjoint lightpaths in G and remove v_i from the current logical graph
  end if
end if

Theorem 12. *Algorithm INCIDENCE-SMART provides a survivable mapping of the edges of a graph* G_L'' *(output of algorithm INCIDENCE-SMART) that contains the given logical graph* G_L.

Proof. *Consider any cut* $(S, \overline{S})$ *in* G_L. *Let S be the partition of the cut that does not contain the datum vertex. Consider the vertex v in S that has the highest index in the INC-sequence. Then in the current graph at the step when v is considered by the algorithm it will not be adjacent to any vertex in S. So, according to the algorithm v will be connected to at least two vertices in* $\overline{S}$, *and the corresponding edges connecting S and* $\overline{S}$ *are mapped into disjoint lightpaths, guaranteeing that at least one of these edges will remain in the cut after a single edge failure in G and satisfying the condition of* Theorem 7. *Since this is true for all cuts, the mappings generated by the algorithm are survivable.* □

We now draw attention to a shortcoming of the algorithmic framework *CUTSET-SMART*. Algorithm *CIRCUIT-SMART* would not require any additional edges to be added to the logical graph if no new edges (protection edges) are added in steps 1–3 of this algorithm. This is not the case with algorithm *CUTSET-SMART*. This algorithm requires protection edges to be added to all unmapped branches. So, *CIRCUIT-SMART* guarantees a survivable mapping of the given logical graph, if steps 1–3 do not require any new edges to be added. On the other hand, *CUTSET-SMART* guarantees a survivable mapping of the graph obtained by contracting the unmapped branches in the logical graph (Theorem 9 (b)), if steps 1–3 of this algorithm do not require any new edges to be added.

The question now arises if it is possible to obtain a generalized version of *CUTSET-SMART* that does not have this limitation. Next we address this question and provide an affirmative answer.

7. Generalized circuit/cutset cover sequences

An ordered sequence $B(c_1)$, $B(c_2)$, …, $B(c_k)$ is a **generalized circuit cover sequence**.

1. if this sequence is a circuit cover sequence, and
2. for every unmapped chord c_i, $B(c_i)\bigcap S(c_j) = S(c_j)$, where j is the largest index such that $B(c_i)\bigcap S(c_j) \neq \emptyset$. In this case, we say that the unmapped chord c_i is **covered** by the chord c_j. We also say that chord c_j covers itself.

An ordered sequence $Q(b_1)$, $Q(b_2)$, …, $Q(b_k)$ is a **generalized cutset cover sequence**.

1. if this sequence is a cutset cover sequence, and
2. for every unmapped branch b_i, $Q(b_i)\bigcap\widehat{S}(b_j) = \widehat{S}(b_j)$, where j is the largest index such that $Q(b_i)\bigcap\widehat{S}(b_j) \neq \emptyset$. In this case we say that the unmapped branch b_i is **covered** by the branch b_j. We also say that branch b_j covers itself.

Given a generalized circuit cover sequence $B(c_1)$, $B(c_2)$, …, $B(c_k)$. We define the set $B - Cover(c_i)$, for each $i = 1, 2, \ldots, k$ as the set of all chords (including itself) covered by the chord c_i. The B-cover sets define a partition of the chords with respect to the given spanning tree. If we arrange the rows of the f-circuit matrix with respect to the spanning tree to correspond to the sets $B - Cover(c_1)$, $B - Cover(c_2)$, …, $B - Cover(c_k)$ in that order and arrange the columns to correspond to the sets $B - Cover(c_1)$, $B - Cover(c_2)$, …, $B - Cover(c_k)$, $S(c_1)$, $S(c_2)$, …, $S(c_k)$ then the f-circuit matrix will have the form shown in (7). In (7), I stands for a matrix of all 1's, 0 is a matrix of 0's and U refers to the unit matrix of appropriate size. Also, $B(c_i)$ stands for $B - Cover(c_i)$.

$B(c_1)$	$B(c_2)$	$B(c_3)$	…	$B(c_{k-1})$	$B(c_k)$	$S(c_1)$	$S(c_2)$	$S(c_3)$	…	$S(c_{k-1})$	$S(c_k)$
U	0	0	…	0	0	I…	0…	0…	…	0…	0…
0	U	0	…	0	0	×…	I…	0…	…	0…	0…
0	0	U	…	0	0	×…	×…	I…	…	0…	0…
…	…	…	…	…	…	…	…	…	…	…	…
0	0	0	…	U	0	××	××	××	××	××	I0
0	0	0	…	0	U	××	××	××	××	××	×I

(7)

Similarly, given a generalized cutset cover sequence $Q(b_1)$, $Q(b_2)$, …, $Q(b_k)$, we define the set $Q - Cover(b_i)$ for each $i = 1, 2, \ldots, k$ as the set of all branches (including itself) covered by the branch b_i. The Q-cover sets define a partition of the branches of the given spanning tree. If we arrange the rows of the f-cutset matrix to correspond to the sets $Q - Cover(b_1)$, $Q - Cover(b_2)$, …, $Q - Cover(b_k)$ in that order and arrange the columns to correspond to the sets $Q - Cover(b_1), Q - Cover(b_2), \ldots, Q - Cover(b_k), \widehat{S}(b_1), \widehat{S}(b_2), \ldots, \widehat{S}(b_k)$, then the f-cutset matrix will have the form shown in (8). In (8), I stands for a matrix of all 1's, 0 is a matrix of 0's and U refers to the unit matrix of appropriate size. Also $Q(b_i)$ stands for $Q - Cover(b_i)$.

$Q(b_1)$	$Q(b_2)$	$Q(b_3)$	…	$Q(b_{k-1})$	$Q(b_k)$	$\hat{S}(b_1)$	$\hat{S}(b_2)$	$\hat{S}(b_3)$	…	$\hat{S}(b_{k-1})$	$\hat{S}(b_k)$
U	0	0	…	0	0	I…	0…	0…	…	0…	0…
0	U	0	…	0	0	×…	I…	0…	…	0…	0…
0	0	U	…	0	0	×…	×…	I…	…	0…	0…
…	…	…	…	…	…	…	…	…	…	…	…
0	0	0	…	U	0	××	××	××	××	××	I0
0	0	0	…	0	U	××	××	××	××	××	×I

(8)

8. Generalized circuit/cutset cover sequence algorithm

Given a cutset cover sequence $Q(b_1)$, $Q(b_2)$, …, $Q(b_k)$ of length k with respect to a spanning tree of a graph. We now present an algorithm GEN-CUTSET-COVER to construct a generalized cutset cover sequence. We use a vector of size n, *GEN_CUTSET_SEQ*, to denote the current cutset cover sequence so that initially.

Algorithm 6
Algorithm GEN-CUTSET-COVER

1: Pick a branch $b_x \in$ *UNMAP*
2: Remove b_x from *UNMAP*
3: Let j be the largest index such that *GEN_CUTSET_SEQ*(j) $= b_y$ and $Q(b_{x})\bigcap \hat{S}(b_{y})\neq \emptyset$
4: **if** $Q (b_{x})\bigcap\hat{S}(b_{y})\neq \hat{S}(b_{y})$ **then**
5: *GEN_CUTSET_SEQ*(j) $\leftarrow b_x$
6: *GEN_CUTSET_SEQ*(r) $\leftarrow$ *GEN_CUTSET_SEQ*($r-1$), $j+1 \leq r \leq n-1$,
7: $\hat{S}(b_{x})\leftarrow Q(b_{x})\bigcap \hat{S}(b_{y})$,
8: $\hat{S}(b_{y})\leftarrow \hat{S}(b_{y}) - \hat{S}(b_{x})$
9: $Q - Cover(b_x) \leftarrow \{b_x\}$
10: **Note:** The *GEN_CUTSET_SEQ* has been modified as $[b_1, b_2, \ldots, b_{j-1}, b_x, b_j, \ldots, b_k, 0, 0, \ldots, 0]$
11: **end if**
12: Add b_x to $Q - Cover(b_y)$
13: Repeat above steps until *UNMAP* is empty

GEN_CUTSET_SEQ(i) $= b_i$, $1 \leq i \leq k$, and.
GEN_CUTSET_SEQ(i) $= 0$, $k+1 \leq i \leq n-1$.

The set *UNMAP* will denote the set of branches that are not in the cutset cover sequence. So, initially *UNMAP* $= \{b_{k+1}, b_{k+2}, \ldots, b_{n-1}\}$. As before, $Q - Cover(b_i)$ will denote the set of all branches covered by branch b_i. So initially, $Q - Cover(b_i) = b_i$.

We first give an informal description of the algorithm *GEN_CUTSET_COVER*. At a general step, the algorithm picks an unmapped branch b_x. Let j be the largest index such that *GEN_CUTSET_SEQ*(j) $= b_y$ and $Q(b_x)\bigcap\widehat{S}(b_y) \neq \emptyset$.

- If $Q(b_x)\bigcap\widehat{s}(b_y) \neq \widehat{S}(b_y)$, then the algorithm inserts b_x as the jth element in *GEN_CUTSET_SEQ* and increments the positions of all subsequent elements by one. The algorithm also sets $\widehat{S}(b_x)$ to $Q(b_x)\bigcap\widehat{S}(b_y)$. For example, if the current cutset cover sequence is b_1, b_2, …, b_{j-1}, b_y, b_{j+1}, …, b_k then the new sequence is b_1, b_2, …, b_{j-1}, b_x, b_y, b_{j+1}, …, b_k.
- Otherwise, that is, if $Q(b_x)\bigcap\widehat{S}(b_y) = \widehat{S}(b_y)$, then the algorithm adds b_x to $Q - Cover(b_y)$.

Theorem 13. *The cutset cover sequence at the termination of Algorithm GEN_CUTSET_COVER is a generalized cutset cover sequence.*

Proof. *The algorithm begins with a cutset cover sequence. Steps* 1–11 *in the algorithm guarantee that the sequence continues to be a cutset cover sequence throughout the execution of the algorithm. Also, every mapped branch is in its own Q-Cover set. So we only need to show that an unmapped branch, once it becomes a member of some* $Q - Cover(b_i)$, *does not leave* $Q - Cover(b_i)$. *Suppose that* b_x *is added to* $Q - Cover(b_y)$ *(because of step* 12 *in the algorithm). At this point* $Q(b_x)\bigcap\widehat{S}(b_y) = \widehat{S}(b_y)$ *and j is the largest index such that GEN_CUTSET_SEQ(j)* $= b_y$. *At the end of every step in the algorithm,* b_y *remains in position j or in a higher position. This guarantees that* b_y *continues to be in the highest position in GEN_CUTSET_SEQ such that the intersection of corresponding* $\widehat{S}(b_y)$ *with* $Q(b_x)$ *is* $\widehat{S}(b_y)$. *So* b_x *continues to be in* $Q - Cover(b_y)$. *Thus, at the end of the algorithm, every branch is in some* $Q - Cover(b_i)$ *and no branch is in more than one* $Q - Cover(b_i)$ *set guaranteeing that the algorithm produces a generalized cutset cover sequence.* □

9. Algorithm GEN-CUTSET-SMART

Given a spanning tree of a logical graph, we now present a generalized version of *CUTSET-SMART* that does not require addition of protection edges to the unmapped branches. This algorithm does not have step 5 of *CUTSET-SMART* (Algorithm 3) and has a modified version of steps 1–4 of *CUTSET-SMART*.

Algorithm 7
Algorithm GEN-CUTSET-SMART

1: Starting with any cutset cover sequence, generate a generalized cutset cover sequence using algorithm *GEN-CUTSET-COVER*. Let this sequence be $Q(b_1)$, $Q(b_2)$, …, $Q(b_k)$
2: **for** each b_i, $i = 1, 2, \ldots, k$ **do**
3: Map a maximum subset of edges in $Q - Cover(b_i) \bigcup \widehat{S}(b_i)$ into disjoint lightpaths in G.
4: To all other edges in $Q - Cover(b_i) \bigcup \widehat{S}(b_i)$, add protection edges and map each edge and its protection edge into disjoint lightpaths in G [31].
5: **end for**

We now prove the correctness of algorithm *GEN-CUTSET-SMART*. In Theorem 14, $G_L^{'}$ refers to the original logical topology G_L augmented with protection edges added as in steps 2–5 of algorithm *GEN-CUTSET-SMART*.

Theorem 14. *The graph $G_L^{'}$ with the mappings generated by algorithm GEN-CUTSET-SMART is survivable.*

Proof. *Consider a cut in $G_L^{'}$. If this cut has a protection edge, then this edge and the corresponding edge in G_L are mapped by the algorithm into disjoint lightpaths in the physical topology. Hence one of them will remain in the cut after a single physical edge failure, thereby satisfying the condition in Theorem 7.*

If there is no protection edge in the cut, then consider the branch b_i in the cut that has the highest index in the generalized cutset cover sequence. If this cut has a branch $b_k \in Q - Cover(b_i)$ with $i \neq k$, then both b_i and b_k are mapped by the algorithm into edge-disjoint paths in the physical topology. Hence one of them will remain in the cut after a single physical edge failure, thereby satisfying the condition of Theorem 7.

If the cut has no branch $b_k \in Q - Cover(b_i)$ with $i \neq k$, then by Theorem 4 (b) the set $\widehat{S}(b_j)$ will be in the cut. Since the branch b_j and the edges in the set $\widehat{S}(b_j)$ are mapped into disjoint manner by the algorithm, the cut will contain at least one edge after a physical edge failure, thereby satisfying the condition of Theorem 7.

Thus in all cases, the algorithm generates a survivable mapping of the graph $G_L^{'}$. □

A generalized version of *CUTSET-SMART-SIMPLIFIED* starting from *GEN-CUTSET-SMART* can be designed and is given in algorithm 8. Algorithm *CIRCUIT-SMART* will remain the same whether we use a circuit cover sequence or a generalized circuit cover sequence. *GEN-CIRCUIT-SMART* will refer to *CIRCUIT-SMART* that uses a generalized circuit cover sequence.

Algorithm 8
Algorithm GEN-CUTSET-SMART-SIMPLIFIED

1: Starting with any cutset cover sequence, generate a generalized cutset cover sequence using algorithm *GEN-CUTSET-COVER*(Algorithm 6). Let this sequence be $Q(b_1)$, $Q(b_2)$, …, $Q(b_k)$.
2: **for** each b_i, $i = 1, 2, \ldots, k$ **do**
3: Pick a chord c_r in set $\widehat{S}(b_i)$. Map a maximum subset of edges in $Q - Cover(b_i) \bigcup c_r$ into edge disjoint lightpaths in G [31]
4: To all other edges in $Q - Cover(b_i) \bigcup c_r$, add protection edges and map each edge and its protection edge into disjoint lightpaths in G
5: **end for**

10. Primal Meets Dual

In this section we first show an interesting relationship between generalized circuit and generalized cutset cover sequences, using the relationship given in Theorem 3. Using this result we then show that the distinction between the primal method (based on circuits) and the dual method (based on cutsets) disappears if these methods are based on generalized circuit and cutset cover sequences.

Let Q_f denote the fundamental cutset matrix with respect to a given generalized cutset cover sequence $Q(b_1)$, $Q(b_2)$, …, $Q(b_k)$. Recall that $Q - Cover(b_i)$ refers to the set of branches covered by the branch b_i in the cover sequence. Note that all branches in $Q - Cover(b_i)$ except b_i are unmapped branches (that is, those branches that are not in the given cover sequence). If we arrange the rows of the *f*-cutset matrix Q_f to correspond to the sets $Q - Cover(b_1)$, $Q - Cover(b_2)$, …, $Q - Cover(b_k)$ in that order and arrange the columns to correspond to the sets $Q - Cover(b_1), Q - Cover(b_2), \ldots, Q - Cover(b_k), \widehat{S}(b_1), \widehat{S}(b_2), \ldots, \widehat{S}(b_k)$, then the submatrix Q_{fc}^t of Q_f whose columns are arranged in the order $\widehat{S}(b_1), \widehat{S}(b_2), \ldots, \widehat{S}(b_k)$, will have the form shown in (9) (see also equation (8)).

Note that for each b_r in the given generalized cutset cover sequence, the rows of the matrix I in the columns corresponding to $\widehat{S}(b_r)$ correspond to the branches in $Q - Cover(b_r)$.

Consider now the fundamental circuit matrix B_f. Then the submatrix B_{ft} with its columns arranged to correspond to the sets $Q - Cover(b_k)$, $Q - Cover(b_{k-1})$, …, $Q - Cover(b_1)$ in that order and the rows arranged to correspond to the sets $\widehat{S}(b_k), \widehat{S}(b_{k-1}), \ldots, \widehat{S}(b_1)$, then in view of the relationship $B_{ft} = Q_{fc}^t$, in Theorem 3, the matrix B_{ft} will have exactly the same structure as in (10) (see also equation (7)), where for the sake of convenience, $\widehat{Q}(b_i)$ stands for $Q - Cover(b_i)$. This leads to Theorem 15.

$$Q_{fc} = \begin{array}{c} \begin{array}{ccccc} \hat{S}(b_1) & \hat{S}(b_2) & \ldots & \hat{S}(b_{k-1}) & \hat{S}(b_k) \end{array} \\ \begin{bmatrix} I & 0 & \ldots & 0 & 0 \\ X & I & \ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ X & X & \ldots & I & 0 \\ X & X & \ldots & X & I \end{bmatrix} \end{array} \tag{9}$$

$$B_{ft} = \begin{array}{c} \begin{array}{ccccc} \hat{Q}(b_k) & \hat{Q}(b_{k-1}) & \ldots & \hat{Q}(b_2) & \hat{Q}(b_1) \end{array} \\ \begin{bmatrix} I & 0 & \ldots & 0 & 0 \\ X & I & \ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ X & X & \ldots & I & 0 \\ X & X & \ldots & X & I \end{bmatrix} \end{array} \tag{10}$$

Theorem 15. *Given a generalized cutset cover sequence $Q(b_1)$, $Q(b_2)$, …, $Q(b_k)$, let chords $c_1, c_2, \ldots, c_k$ be selected such that each $c_i \in \widehat{S}(b_j)$, where $j = k - i + 1$. Then the sequence $B(c_1)$, $B(c_2)$, …, $B(c_k)$ is a generalized circuit cover sequence with*

- *$B - Cover(ci) = \widehat{S}(b_{k-i+1})$; and*
- *$S(c_i) = Q - Cover(b_{k-i+1})$.*

Starting from a generalized circuit cover sequence one can also get a generalized cutset cover sequence and a result dual to Theorem 15 is stated in Theorem 16.

Theorem 16. *Given a generalized circuit cover $B(c_1)$, $B(c_2)$, …, $B(c_k)$, let branches $b_1, b_2, \ldots, b_k$ be selected such that each $b_i \in S(c_{k-i+1})$. Then*

- *The sequence $Q(b_1)$, $Q(b_2)$, …, $Q(b_k)$ is a generalized cutset cover sequence;*
- *$Q - Cover(b_i) = S(c_{k-i+1})$; and*
- *$\widehat{S}(b_i) = B - Cover(c_{k-i+1})$.*

In view of Theorems 15 and 16, we get the following.

Theorem 17. *(Primal Meets Dual): If a generalized circuit cover sequence or the corresponding generalized cutset cover sequence is used, then the set of* *edges picked (for disjoint mappings) in each execution of steps 1–4 by algorithm CIRCUIT-SMART corresponds to the set of edges picked (for disjoint mappings) in an execution of steps 1–5 of algorithm GEN-CUTSET-SMART-SIMPLIFIED.*

As an example, consider the B_f matrix in (1). It can be verified that $B(c_2)$, $B(c_6)$, $B(c_3)$, $B(c_4)$ is a generalized circuit cover sequence with.

$B - Cover(c_2) = \{c_2\}$, $S(c_2) = \{b_1, b_2\}$

$B - Cover(c_6) = \{c_6, c_1\}$, $S(c_6) = \{b_3\}$

$B - Cover(c_3) = \{c_3\}$, $S(c_3) = \{b_4\}$

$B - Cover(c_4) = \{c_4, c_5\}$, $S(c_4) = \{b_5\}$

Picking the branches selected as in Theorem 16, we can get the following generalized cutset cover sequence $Q(b_5)$, $Q(b_4)$, $Q(b_3)$, $Q(b_2)$ with.

$Q - Cover(b_5) = \{b_5\}, \widehat{S}(b_5) = \{c_4, c_5\}$

$Q - Cover(b_4) = \{b_4\}, \widehat{S}(b_4) = \{c_3\}$

$Q - Cover(b_3) = \{b_3\}, \widehat{S}(b_3) = \{c_1, c_6\}$

$Q - Cover(b_2) = \{b_1, b_2\}, \widehat{S}(b_2) = \{c_2\}$

An execution of algorithm *CIRCUIT-SMART* will require the disjoint mappings of the following sets:

$\{c_2, b_1, b_2\}$, $\{c_6, b_3\}$, $\{c_3, b_4\}$, $\{c_4, b_5\}$.

If we apply algorithm *GEN-CUTSET-SMART-SIMPLIFIED* by choosing c_4 from $\widehat{S}(b_5)$ and c_6 from $\widehat{S}(b_3)$ then we will select the above sets of edges for disjoint mappings as stated in Theorem 17.

In view of Theorem 17, algorithm *GEN-CUTSET-SMART-SIMPLIFIED* may be viewed as a primal (circuit based) or dual (cutset based) algorithm. It can also be viewed as a primal-dual algorithm if we replace steps 1–5 of *GEN-CUTSET-SMART-SIMPLIFIED* by the following:

(a) Dual Step (Cutset based): if $Q - Cover(b_i)$ has exactly one branch, namely, itself, then pick any chord c_i in $\widehat{S}(b_i)$ and map c_j and b_i into disjoint lightpaths in the physical topology.

(b) Primal Step (Circuit based): if $Q - cover(b_i)$ has more than one branch, then pick any chord c_j in $\widehat{S}(b_i)$, and map c_j and all the branches in $Q - Cover(b_i)$ into disjoint light paths in the physical topology.

We call step (a) above as dual step since it is the same as step 1–5 in *GEN-CUTSET-SMART-SIMPLIFIED*. Step (b) is called the primal step because in view of Theorem 3 it is the same as steps 1–4 in CIRCUIT-SMART with respect to the chord c_i.

11. GEN-SMART: A generalized algorithmic framework for the SLTM problem

In this section we present *GEN-SMART* (Algorithm 9), an algorithmic framework for the survivable logical topology mapping (SLTM) problem. This framework includes as special cases the other SMART-based algorithms discussed in previous sections.

Algorithm 9
Algorithm GEN-SMART

```
Starting with any cutset cover sequence, generate a generalized cutset cover
  sequence of G_L. Let this sequence be Q(b_1), Q(b_2), ..., Q(b_k)
for each i = 1, 2, ..., k do
  Let A ⊆ Ŝ(b_i) and B ⊆ Q − Cover(b_i)
  Map the edges in the set b_i ⋃ A ⋃ B into disjoint lightpaths in G
end for
```

For the sake of simplicity in presentation we ave assumed in the description of *GEN-SMART* that all the edges in the set $A \subseteq \widehat{S}(b_i)$ and $B \subseteq Q - Cover(b_i)$ can be mapped into disjoint paths in G. But this may not always be possible. In such cases, we map a maximum subset of these edges into disjoint paths. To the other edges in this set, we add protection edges and map each edge and its protection edge into disjoint paths in G. Also, if we choose $A = \widehat{S}(b_i)$ and $B = Q - Cover(b_i)$ then *GEN-SMART* becomes the same as *GEN-CUTSET-SMART*. Also, different choices of A and B in *GEN-SMART* lead to different versions of SMART-based algorithms discussed in the previous sections (see Table 1).

Table 1
Special cases of GEN-SMART algorithms.

Choice of A and B	Special case of GEN-SMART
$\lvert A\rvert = 1,\ \lvert B\rvert = 1$	*GEN-CUTSET-SMART-SIMPLIFIED*
$\lvert A\rvert = \widehat{S}(b_i), \lvert B\rvert = 1$	*CUTSET-SMART*
$\lvert A\rvert = 1,\ \lvert B\rvert = Q - Cover(b_i)$	*CIRCUIT-SMART*
$\lvert A\rvert = \widehat{S}(b_i), \lvert B\rvert = Q - Cover(b_i)$	*GEN-CUTSET-SMART*

Some observations on the different versions of GEN-SMART are now in order.

- GEN-CUTSET-SIMPLIFIED and CUTSET-SMART do not guarantee survivability even against a single physical link failure unless protection edges are added to the unmapped branches (those branches not in B).
- CIRCUIT-SMART and GEN-CUTSET-SMART guarantee survivability against a single physical link failure. Moreover, CIRCUIT-SMART allows unmapped chords (those chords not in A) to be mapped arbitrarily. Both these algorithms have higher potential to provide survivability against multiple physical link failures because in both these algorithms all the edges in $\widehat{S}(b_i)$ are mapped disjointly.

In the next section we provide an analytical evaluation of the extent to which these algorithms provide survivability against multiple failures.

12. Robustness of survivable logical topology mapping algorithms

In this section we first define the concept of robustness of an algorithm that is a measure of the ability to provide survivability against multiple physical failures.

Given a logical topology G_L and a physical topology G, the robustness $\beta(A, r)$ of a logical topology mapping algorithm A with respect to G and G_L is defined as the ratio of the number of cuts of G_L that are guaranteed to be protected by algorithm A against r physical link failures to the total number of cuts in G_L.

We now proceed to evaluate $\beta(A, r)$ for different algorithms. In the following A_1, A_2, A_3 and A_4 denote algorithms *CUTSET-SMART-SIMPLIFIED*, *CUTSET-SMART*, *CIRCUIT-SMART* and *GEN-CUTSET-SMART*, respectively.

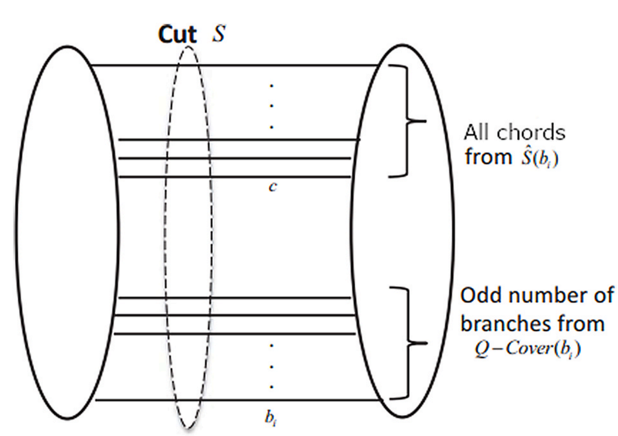


Fig. 5. A cut S.

Given a generalized cutset cover sequence $Q(b_1)$, $Q(b_2)$, …, $Q(b_k)$. Let us first partition all cuts in G_L into the sets $Q_1, Q_2, \ldots, Q_k$ where Q_i is the set of all cuts that contain at least one branch from the set $Q - Cover(b_i)$ and no branch from any set $Q - Cover(b_j)$, $j > i$. Note that this partition is well defined since every cut must have at least one branch.

Consider now a cut $S \in Q_i$. Assume that S contains p branches from Q_i.

In view of Theorem 1(a), the cut S will have the form in Fig. 5 if S has an odd number p of branches from the set $Q - Cover(b_i)$. Note that if p is even then none of the chords in $\widehat{S}(b_i)$ will be in S. Then, the number of edges mapped disjointly by the different Algorithms A_1, A_2, A_3, A_4 are:

- Algorithm A_1 maps edges b_i and a chord c in $\widehat{S}(b_i)$ disjointly, if S contains b_i.
- Algorithm A_2 maps b_i and all edges in $\widehat{S}(b_i)$ disjointly, if S contains b_i.
- Algorithm A_3 maps all the p branches and a chord c in $\widehat{S}(b_i)$.
- Algorithm A_4 maps all the p branches and all the chords in $\widehat{S}(b_i)$.

Thus we have the following:

- Algorithm A_1 protects S against one physical link failure, if S contains b_i.
- Algorithm A_2 protects S against at least $|\widehat{S}(b_i)|$ physical link failures, if S contains b_i.
- Algorithm A_3 protects S against at least p physical link failures.
- Algorithm A_4 protects S against at least $p + |\widehat{S}(b_i)| - 1$ physical link failures.

Since $p \geq 1$, we can restate the last statement as:

- Algorithm A_4 protects S against at least $|\widehat{S}(b_i)|$ physical link failures.

Let $h_i = |Q - Cover(b_i)|, g_i = |\widehat{S}(b_i)|$.
$h = \min |h_i|$ and $g = \min |g_i|$.
Also, let $N_i = h_1 + h_2 + \cdots + h_i$.

12.1. Robustness of algorithm A_1

Algorithm A_1 will protect against a single physical failure all cuts from each Q_i that has an odd number of branches from the set $Q - Cover$ (b_i) and contains branch b_i. This number is equal to.

= (Number of combinations of branches from the sets $Q - Cover(b_k)$, $k = 1, 2, \ldots, i - 1$) × (Number of combinations of odd number of branches from the set $Q - Cover(b_i)$ that contains b_i).

$= 2^{N_{i-1}} \times 2^{h_{i-2}}$

$= 2^{N_i}/4$.

Since the number of cuts in G_L is $2^{n-1} - 1$, where n is the number of nodes in G_L, and $n - 1 = h_1 + h_2 + \cdots + h_k$, we get

$$\beta(A_1, 1) \geq 1 \Big/ 4\left(\sum_{i=1}^{k} 2^{N_i}\right) \Big/ \left(2^{n-1} - 1\right) \tag{11}$$

Note that if $p \geq 2$, $\beta(A, p) \geq 0$, since there is no guarantee that algorithm A_1 will protect any cut if 2 or more physical failures occur.

12.2. Robustness of algorithm A_2

Algorithm A_2 will protect against g_i physical failures all cuts from each Q_i that have an odd number of branches from the set $Q - Cover(b_i)$ and contain branch b_i. This follows from the fact that each such cut will have b_i and all edges in $\widehat{S}(b_i)$ that are mapped disjointly.

So,

$$\beta(A_2, g_i) \geq 1 \Big/ 4\left(\sum_{i=1}^{k} 2^{N_i}\right) \Big/ \left(2^{n-1} - 1\right) \tag{12}$$

12.3. Robustness of algorithm A_3

Algorithm A_3 will protect against at least p physical failures all cuts from each Q_i that has an odd number p of branches from the set $Q - Cover(b_i)$. This follows from the fact that each such cut will have p branches and at least one chord c in $\widehat{S}(b_i)$ that are mapped disjointly.

This number is equal to

$$\beta(A_3, p) \geq \left(\sum_{i=1}^{k} 2^{N_{i-1}} \sum_{\text{odd } q \geq p}^{h_i} C(h_i, q)\right) \Big/ \left(2^{n-1} - 1\right), \quad \text{for odd} \quad p \geq 1 \tag{13}$$

where $C(h_i, q)$ is the number of q-combinations of h_i elements.

12.4. Robustness of algorithm A_4

Algorithm A_4 will protect against at least $|\widehat{S}(b_i)|$ physical failures all cuts from each Q_i that have an odd number of branches from the set $Q - Cover(b_i)$. This follows from the fact that each such cut will have at least one branch and all the chords in $\widehat{S}(b_i)$ that are mapped disjointly.

This number is equal to (Number of combinations of branches from the sets $Q - Cover(b_k)$, $k = 1, 2, \ldots, i - 1$) × (Number of combinations of p branches from the set $Q - Cover(b_i)$)

$= 2^{N_{i-1}} \times 2^{h_i - 1}$
$= 2^{N_i}/2$

So,

$$\beta(A, p) \geq 1 \Big/ 2\left(\sum_{i=1}^{k} 2^{N_i}\right) \Big/ \left(2^{n-1} - 1\right) \tag{14}$$

Let $SUM = (\sum_{i=1}^{k} 2^{N_i})/(2^{n-1} - 1)$. Then, we can write the robustness of algorithms A_1, A_2, A_4 as in the following Theorem.

Theorem 18.

$\beta(A_1, 1) \geq 1/4SUM$
$\beta(A_2, g) \geq 1/4SUM$
$\beta(A_4, g) \geq 1/2SUM.$

The value of SUM depends on the choice of generalized cutset cover sequence selected. Note that SUM is at most 2. flushleft

Let $SUM = \left(\sum_{i=1}^{k} 2^{N_i}\right)/(2^{n-1} - 1)$.

Then we can rewrite (11), (12), (14) as

$$\beta(A_1, 1) \geq 1/4 \ SUM \tag{15}$$

$$\beta(A_2, g) \geq 1/4 \ SUM \tag{16}$$

$$\beta(A_4, g) \geq 1/2 \ SUM \tag{17}$$

Depending on the length of the generalized cutset cover sequence, the sizes of h_i's and g_i's, the location of physical link failures and the mappings used, the number of protected cuts could be much larger. The higher the value of $\beta(A, r)$ the higher will be the probability that algorithm A will protect G_L from any set of r physical link failures.

13. Summary, comparative evaluation and applications beyond IP-over-WDM optical networks

In this section we present extensive simulation results comparing the different algorithms developed in this paper. The comparisons are in terms of three metrics: execution time, number of protection edges added, and robustness (that is, the ability to provide survivability against multiple failures). We also point to applications of our work in areas beyond IP-over-WDM optical networks, in particular, design of survivable multi-layer interdependent networks.

13.1. Execution time and number of protection edges added

To compare the performance of the CIRCUIT-SMART, CUTSET-SMART-SIMPLIFIED and GEN-CUTSET-SMART-SIMPLIFIED simulation studies were conducted using VC++ 8.0. For simulation studies, random logical topologies with varying number of nodes and degrees were generated. The physical topologies were regular topologies with degree 4, constructed using a procedure originally given by Harary and described in Ref. [30]. The number of nodes in the physical topologies was set to 50, 100, and 200 nodes ($|N|$). The logical topologies were generated randomly with average degrees 2.5, 3.0, 3.5 and 4.0. The nodes in the logical topologies were a subset of the physical nodes and number of logical nodes in the logical topology was set to $0.75 \times |N|$. For each case, 10 physical and 10 logical topologies were generated, providing a total of 100 logical-physical topology pairs for comparison. The topologies were subjected to further processing, only if the topologies met the connectivity requirements.

To find the maximum number of logical links that could be mapped in a mutually disjoint manner, a procedure described in Ref. [31] was used. To find mutually disjoint mappings of a pair of logical links, the algorithm given in Ref. [32] was used. Fundamental circuits and cutsets were found using procedures given in Ref. [30] and were part of the preprocessing phase. The survivability of a logical topology was tested by picking a physical link, removing all the logical links which used this physical link in their mapping, and checking if the resulting logical topology was connected. This test was repeated for every physical link.

The statistics of interest were protection capacity (measured as average number of protection links added to a logical topology to make it survivable) and the execution time of the algorithms.

We now make some general observations on the performance of these algorithms based on the trends that we noticed during the simulations as in Table 2 and Table 3. The results in these two tables confirm our expectation that GEN-CUTSET-SMART-SIMPLIFIED will perform better than CIRCUIT-SMART and CUTSET-SMART-SIMPLIFIED (when started with a circuit or a cutset sequence) in terms of number of additional protection edges to be added. In terms of execution time, CUTSET-SMART-SIMPLIFIED performs significantly better than CIRCUIT-SMART and GEN-CUTSET-SMART-SIMPLIFIED when the logical topologies are sparse but at the same time requires significantly more protection edges. However, for dense logical topologies the difference in execution times is much smaller. Note that CIRCUIT-SMART and CUTSET-SMART-SIMPLIFIED were implemented with arbitrary circuit and cutset cover sequences. GEN-CUTSET-SMART-SIMPLIFIED was implemented with a generalized cutset cover sequence.

Table 2
Ave. No. of Protection Edges and Execution Time (Number of nodes = 50, Physical degree = 4.)

Logical Degree	Algorithm	Protection Edges	Execution Time (sec)
2.5	CIRCUIT-SMART	15.99	0.543 44
	CUTSET-SMART-SIMPLIFIED	26.23	0.071 91
	GEN-CUTSET-SMART-SIMPLIFIED	12.93	0.508 66
3.0	CIRCUIT-SMART	8.86	0.394 54
	CUTSET-SMART-SIMPLIFIED	20.86	0.085 65
	GEN-CUTSET-SMART-SIMPLIFIED	7.48	0.403 93
3.5	CIRCUIT-SMART	5.98	0.327 5
	CUTSET-SMART-SIMPLIFIED	16.1	0.097 35
	GEN-CUTSET-SMART-SIMPLIFIED	5.5	0.312 49
4.0	CIRCUIT-SMART	3.92	0.259 4
	CUTSET-SMART-SIMPLIFIED	12.66	0.107 01
	GEN-CUTSET-SMART-SIMPLIFIED	4.74	0.281 24

Table 3
Ave. No. of Protection Edges and Execution Time (Number of nodes = 100, Physical degree = 4.)

Logical Degree	Algorithm	Protection Edges	Execution Time (sec)
2.5	CIRCUIT-SMART	41.59	4.929 08
	CUTSET-SMART-SIMPLIFIED	56.57	0.309 11
	GEN-CUTSET-SMART-SIMPLIFIED	35.8	4.762 46
3.0	CIRCUIT-SMART	25.14	3.622 06
	CUTSET-SMART-SIMPLIFIED	44.04	0.392 99
	GEN-CUTSET-SMART-SIMPLIFIED	20.4	3.416 03
3.5	CIRCUIT-SMART	14.07	2.504 07
	CUTSET-SMART-SIMPLIFIED	36.08	0.461 03
	GEN-CUTSET-SMART-SIMPLIFIED	11.48	2.440 2
4.0	CIRCUIT-SMART	10.67	2.251 27
	CUTSET-SMART-SIMPLIFIED	26.72	0.527 82
	GEN-CUTSET-SMART-SIMPLIFIED	9.44	2.158 48

It was shown in Ref. [3] that CUTSET-SMART-SIMPLIFIED and INCIDENCE-SMART are comparable with respect to execution times. But INCIDENCE-SMART is found to be better in terms of the number of protection edges needed.

13.2. Multi-failure survivability

In this section we present simulation results that provide a comparative evaluation of the algorithms CUTSET-SMART-SIMPLIFIED, CUTSET-SMART, CIRCUIT-SMART and GEN-CUTSET-SMART which are denoted as A_1, A_2, A_3, and A_4, respectively.

To compare the performance of CUTSET-SMART-SIMPLIFIED, CIRCUIT-SMART, CUTSET-SMART, and GEN-CUTSET-SMART with respect to their ability to provide multiple failure survivability simulation studies were conducted using LEMON (Library for Efficient Modeling and Optimization in Networks) [33] and G++ under Linux system. The physical and logical topologies were regular topologies with connectivity equal to 3, 4, and 5 constructed using a procedure originally given by Harary and described in Ref. [30]. The number of nodes in the physical topologies was set to 50, 60, 70, 80, 90, and 100 nodes. The nodes in logical topologies were a subset of the physical nodes and the number of nodes in a logical topology was set to 50% of the nodes in the corresponding physical topology.

For each combination of (topology connectivity, number of nodes in physical topology, number of physical link failures), 100 physical and corresponding logical topology pairs were generated and tested against 4 algorithms described in the previous section. Given k-connected physical and logical topologies, the survivability of the G_L under multiple (2 to $k-1$) physical link failures is determined by the number of G_L's which remain connected against physical link failures. Our simulation enumerated all possible combinations of physical link failures and evaluated how many G_L's could remain connected. The success rate in each case is calculated.

First a spanning tree on a logical topology was generated and the fundamental circuits and cutsets with respect to the spanning tree were found. The generalized cutset cover sequence was generated using the algorithm 2. With the information of the fundamental cutsets, the $Q-Cover(b_i)$ and $\widehat{S}(b_i)$ sets were generated as shown in (2) and (3). Then we applied the four algorithms (CUTSET-SMART-SIMPLIFIED, CIRCUIT-SMART, CUTSET-SMART, and GEN-CUTSET-SMART) and mapped

Table 4
Success rate for 3-connected physical and logical topologies.

3-conn Algorithms \ Failures	50 nodes 1	50 nodes 2	60 nodes 1	60 nodes 2	70 nodes 1	70 nodes 2
A_1	92.173	71.857	89.711	65.294	89.429	64.338
A_2	92.987	73.701	90.533	67.080	90.371	66.024
A_3	100	85.367	100	83.775	100	82.263
A_4	100	86.426	100	84.406	100	83.375
3-conn Algorithms \ Failures	**80 nodes 1**	**80 nodes 2**	**90 nodes 1**	**90 nodes 2**	**100 nodes 1**	**100 nodes 2**
A_1	87.617	57.744	86.570	55.356	84.427	52.313
A_2	88.700	59.710	87.963	57.2	85.853	54.405
A_3	100	78.811	100	78.377	100	76.149
A_4	100	79.913	100	79.367	100	77.073

Table 5
Success rate for 4-connected physical and logical topologies.

4-conn Algorithms \ Failures	50 nodes 2	50 nodes 3	60 nodes 2	60 nodes 3	70 nodes 2	70 nodes 3
A_1	94.709	85.841	93.907	84.533	93.655	81.356
A_2	95.975	88.679	95.272	86.979	94.841	84.513
A_3	96.646	88.262	95.950	87.549	95.383	85.219
A_4	97.367	90.263	96.665	89.159	96.235	86.984
4-conn Algorithms \ Failures	**80 nodes 2**	**80 nodes 3**	**90 nodes 2**	**90 nodes 3**	**100 nodes 2**	**100 nodes 3**
A_1	92.381	80.498	91.575	78.445	91.000	76.815
A_2	94.018	83.343	93.373	81.564	93.043	79.780
A_3	94.801	83.473	93.983	81.802	93.466	79.700
A_4	95.639	85.396	95.018	83.819	94.41	81.582

Table 6
Success rate for 5-connected physical and logical topologies.

5-conn Algorithms \ Failures	50 nodes 2	50 nodes 3	60 nodes 2	60 nodes 3	70 nodes 2	70 nodes 3
A_1	99.764	99.450	99.785	99.366	99.809	99.246
A_2	99.912	99.653	99.880	99.634	99.888	99.583
A_3	99.877	99.617	99.869	99.541	99.867	99.473
A_4	99.956	99.810	99.935	99.771	99.937	99.746
5-conn Algorithms \ Failures	**80 nodes 2**	**80 nodes 3**	**90 nodes 2**	**90 nodes 3**	**100 nodes 2**	**100 nodes 3**
A_1	99.772	99.231	99.668	99.184	99.674	99.089
A_2	99.858	99.557	99.785	99.510	99.787	99.507
A_3	99.848	99.827	99.827	99.437	99.804	99.363
A_4	99.916	99.915	99.915	99.725	99.899	99.654

maximal number of edges disjointly in $b_i \bigcup A \bigcup B$. If the disjoint mappings for some of the edges in $b_i \bigcup A \bigcup B$ do not exist, a parallel edge is added to the logical topology and the newly added edge is mapped disjointly with the original edge. At the end of the procedure, the unmapped logical edges were randomly mapped, which could increase the chance of survivability for the logical mapping.

The simulation results giving the success rate are shown in Tables 4–6. Notice that in Table 4, extra tests for the single failure case in 3-connected physical and logical topologies are presented, which show that CUTSET-SMART and GEN-CUTSET-SMART can guarantee 100% survivability for the logical topology under a single physical link failure, while CUTSET-SMART-SIMPLIFIED and CIRCUIT-SMART can not.

Based on the simulations, we summarize our observations as follows.

- The value of *SUM* is at most 2. This can be reached when each $h_i = 1$. In such cases, (15) and (16) simplify to $\beta(A_1, 1) \geq 1/2$, $\beta(A_2, g) \geq 1/2$. In spite of this low value on the corresponding robustness, algorithms A_1 and A_2 have good ability to provide survivability against multiple physical link failures.
- As expected, A_2 has higher potential to provide survivability against multiple failures compared to A_1.
- As expected, algorithms A_3 and A_4 have higher success rate compared to A_1 and A_2.
- The success rate of all algorithms is higher for higher values of connectivity of physical topologies. This could be due to the availability of a large number of disjoint paths. This calls for future research.

Based on the above results, we can make some general observations.

1. If the number of protection edges is of concern, then we recommend CIRCUIT-SMART.
2. If computation time is of concern, we recommend CUTSET-SMART-SIMPLIFIED.
3. If ability to provide survivability against multiple failures is of interest, then we recommend GEN-CUTSET-SMART.

In conclusion, we wish to note that we have been able to use the power of duality to design cutset-based algorithms that have performance superior or comparable to the circuit-based approach. The algorithmic strategy developed in this paper will also be of interest in designing strategies for the SLTM problems when node failures are allowed [34,35].

13.3. Applications beyond IP-over-WDM optical networks

In this work we have made the implicit assumption that the logical topology has lower connectivity than the physical topology. If the connectivity of the logical topology is much higher than the physical topology, then several alternate routes will become available at the logical layer. Then, one can take advantage of these routes at the logical layer, and in combination with the high resilience at the physical layer provided by our algorithmic strategies, one can achieve much higher resilience of the overall network. Furthermore, though the algorithms developed in this paper have been motivated by the survivable logical topology mapping problem in an IP-over-WDM optical network, they are also of interest in solving sub-problems that arise in the design of survivable inter-dependent multi-layer networks. A number of references pointing to applications of our work in the design of survivable cyber-physical systems may be found in Refs. [36–38] and the references therein. In particular, our strategies could serve as the basis of designing approaches to mitigate cascading failures in interdependent networks.

Credits

Thulasiraman: Conceptualization, Methodology, Formal analysis, Funding acquisition, Research Administration. Lin: Conceptualization, Methodology, Formal analysis, Research Administration. Javed: Conceptualization, Methodology, Formal analysis. Zhou: Conceptualization, Methodology, Formal analysis. Xue: Conceptualization, Methodology, Formal analysis, Funding acquisition.

Declaration of competing interest

None.

References

[1] E. Modiano, A. Narula-Tam, Survivable routing of logical topologies in WDM networks, in: Proc. IEEE INFOCOM, 2001, pp. 348–357.
[2] M. Kurant, P. Thiran, On survivable routing of mesh topologies in IP-over-WDM networks, in: Proc. IEEE INFOCOM, 2005, pp. 1106–1116.
[3] K. Thulasiraman, M. Javed, G. Xue, Circuits/cutsets duality and a unified algorithmic framework for survivable logical topology design in IP-over-WDM optical networks, in: Proc. IEEE INFOCOM, 2009, pp. 1026–1034.
[4] K. Thulasiraman, M. Javed, G. Xue, Primal meets dual: a generalized theory of logical topology survivability in IP-over-WDM optical networks, in: Proc. Int. Conf. on Communication Systems and Networks (COMSNETS), 2010, pp. 128–137.
[5] T. Lin, K. Thulasiraman, G. Xue, Robustness of logical topology mapping problems for survivability under multiple physical links failures in IP-over-WDM optical networks, Proc. COMSNETS (2012) 128–137.
[6] R. Karp, Reducibility among combinatorial problems, in: R. Miller, J. Thatcher, J. Bohlinger (Eds.), Complexity of Computer Computations, Springer US, Boston, MA, 1972, pp. 85–103.
[7] A. Todimala, B. Ramamurthy, A scalable approach for survivable virtual topology routing in optical WDM networks, IEEE J. Sel. Area. Commun. 25 (6) (2007) 63–69.
[8] K. Lee, E. Modiano, H. Lee, Cross-layer survivability in WDM-based networks, IEEE/ACM Trans. Netw. 19 (4) (2011) 1000–1013.
[9] D. Kan, A. Narula-Tam, E. Modiano, Lightpath routing and capacity assignment for survivable IP-over-WDM networks, in: 2009 7th International Workshop on Design of Reliable Communication Networks (DRCN), 2009, pp. 37–44.
[10] T. Lin, Z. Zhou, K. Thulasiraman, Logical topology survivability in IP-over-WDM networks: survivable lightpath routing for maximum logical topology capacity and minimum spare capacity requirements, in: 2011 8th International Workshop on the Design of Reliable Communication Networks (DRCN), 2011, pp. 1–8.
[11] T. Lin, Z. Zhou, K. Thulasiraman, G. Xue, S. Sahni, Unified mathematical programming frameworks for survivable logical topology routing in IP-over-WDM optical networks, IEEE/OSA J. Optic. Commun. Netw. 6 (2) (2014) 190–203.
[12] T. Lin, Z. Zhou, K. Thulasiraman, G. Xue, Optimum logical topology routing in an IP-over-WDM optical network and physical link failure localization: an integrated approach, in: 2013 5th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2013, pp. 82–88.
[13] B. Wu, P.-H. Ho, K. Yeung, Monitoring trail: on fast link failure localization in all-optical WDM mesh networks, J. Lightwave Technol. 27 (18) (2009) 4175–4185.
[14] S. Ahuja, S. Ramasubramanian, M. Krunz, Single-link failure detection in all-optical networks using monitoring cycles and paths, IEEE/ACM Trans. Netw. 17 (4) (2009) 1080–1093.
[15] Z. Zhou, T. Lin, K. Thulasiraman, G. Xue, S. Sahni, Cross-layer network survivability under multiple cross-layer metrics, IEEE/OSA J. Optic. Commun. Netw. 7 (2015) 540–553.
[16] M. Rahman, R. Boutaba, Svne: survivable virtual network embedding algorithms for network virtualization, IEEE Trans. Netw. Serv. Manag. 10 (2) (2013) 105–118.
[17] M. Rahman, R. Boutaba, Survivable network virtualization for single facility node failure: a network flow perspective, Opt. Switch. Netw. 10 (4) (2013) 406–415.
[18] Z. Zhou, T. Lin, K. Thulasiraman, G. Xue, Novel survivable logical topology routing by logical protecting spanning trees in IP-Over-WDM networks, IEEE/ACM Trans. Netw. 25 (3) (2017) 1673–1685.
[19] S. Lee, J. Liu, Y. Chen, An ear-decomposition based approach for survivable routing in WDM networks, in: 19th International Conference on Advanced Information Networking and Applications (AINA'05), vol. 1, 2005, pp. 459–464.
[20] M. Javed, K. Thulasiraman, M. Gaines, G. Xue, Survivability aware routing of logical topologies: on thiran-kurant approach, enhancements and evaluation, in: GLOBECOM - IEEE Global Telecommunications Conference, 2006.
[21] M. Javed, K. Thulasiraman, G. Xue, Lightpaths routing for single link failure survivability in IP-over-WDM networks, J. Commun. Network. 9 (4) (2007) 394–401.
[22] K. Thulasiraman, T. Lin, M. Javed, G. Xue, Logical topology augmentation for guaranteed survivability under multiple failures in IP-over-WDM optical networks, Opt. Switch. Netw. 7 (4) (2010) 206–214.
[23] C. Liu, L. Ruan, A new survivable mapping problem in ip-over-wdm networks, IEEE J. Sel. Area. Commun. 25 (3) (2007) 25–34.
[24] Q. Deng, G. Sasaki, C. Su, Survivable IP-over-WDM: an efficient mathematical programming problem formulation, in: 40th Annual Allerton Conference on Control, Communication and Computing, 2002, pp. 518–527.
[25] Z. Zhou, T. Lin, K. Thulasiraman, Survivable cloud network mapping with multiple failures, in: 2015 IEEE International Conference on Communications (ICC), 2015, pp. 5491–5496.
[26] Z. Zhou, T. Lin, K. Thulasiraman, Survivable cloud network design against multiple failures through protecting spanning trees, J. Lightwave Technol. 35 (2) (2017) 288–298.
[27] R. Cohen, G. Nakibly, Restorable logical topology in the face of no or partial traffic demand knowledge, IEEE/ACM Trans. Netw. 24 (4) (2016) 2074–2085.
[28] M. Swamy, K. Thulasiraman, Graphs, Networks and Algorithms, Wiley, 1981.
[29] R. Ahuja, T. Magnanti, J. Orlin, Network Flows: Theory, Algorithms and Applications, Prentice-Hall, 1993.
[30] K. Thulasiraman, M.N.S. Swamy, Graphs: Theory and Algorithms, John Wiley & Sons, 1992.
[31] J. Kleinberg, Approximation Algorithms for Disjoint Path Problems, Ph.D. thesis, MIT, 1996.
[32] Y. Perl, Y. Shiloach, Finding two disjoint paths between two pairs of vertices in a graph, J. ACM 25 (1) (1978) 1–9.
[33] Library for efficient modeling and optimization in networks (LEMON). http://lemon.cs.elte.hu/trac/lemon.
[34] N. Radics, L. Bajzik, Z. Lakatos, Survivable mapping of virtual topologies for double-node failure, IEEE/ACM Trans. Netw. 23 (6) (2015) 1903–1916, https://doi.org/10.1109/TNET.2014.2345099.
[35] L. Zhang, N. Zhang, Survivable virtual topology design for single-node failure, in: 2011 3rd International Conference on Computer Research and Development, vol. 2, IEEE, 2011, pp. 465–467.
[36] Z. Huang, C. Wang, M. Stojmenovic, A. Nayak, Balancing system survivability and cost of smart grid via modeling cascading failures, IEEE Trans. Emerg. Topic. Comput. 1 (1) (2013) 45–56, https://doi.org/10.1109/TETC.2013.2273079.
[37] L. Xu, Q. Guo, T. Yang, H. Sun, Robust routing optimization for smart grids considering cyber-physical interdependence, IEEE Trans. Smart Grid 10 (5) (2019) 5620–5629, https://doi.org/10.1109/TSG.2018.2888629.
[38] R. Liu, C. Vellaithurai, S.S. Biswas, T.T. Gamage, A.K. Srivastava, Analyzing the cyber-physical impact of cyber events on the power grid, IEEE Trans. Smart Grid 6 (5) (2015) 2444–2453, https://doi.org/10.1109/TSG.2015.2432013.