

# Primal Meets Dual: A Generalized Theory of Logical Topology Survivability in IP-over-WDM Optical Networks

Krishnaiyan Thulasiraman and Muhammad Javed

School of Computer Science  
University of Oklahoma  
Norman, OK, USA  
{thulasi, javed}@ou.edu

Guoliang (Larry) Xue

Department of Computer Science and Engineering  
Arizona State University  
Tempe, AZ, USA  
xue@asu.edu

**Abstract**—The survivable logical topology mapping (SLTM) problem in an IP-over-WDM optical network is to map each link  $(u, v)$  in the logical topology (at the IP layer) into a lightpath between the nodes  $u$  and  $v$  in the physical topology (at the optical layer) such that failure of a physical link does not cause the logical topology to become disconnected. It is assumed that both the physical and logical topologies are 2-edge connected. For this problem Kurant and Thiran [12] presented an algorithmic framework called SMART that involves successively contracting circuits in the logical topology and mapping the logical links in the circuits into edge disjoint lightpaths in the physical topology. In a recent work [21] we presented a dual framework involving cutsets and showed that both these frameworks possess the same algorithmic structure. Algorithms CIRCUIT-SMART, CUTSET-SMART, CUTSET-SMART-SIMPLIFIED and INCIDENCE-SMART were also presented in [21]. Effectiveness of both these frameworks as well as their robustness in providing survivability against multiple failures depends on the lengths of the cutset cover and circuit cover sequences on which they are based. To improve their effectiveness and robustness, in this paper we first introduce the concept of generalized cutset cover and generalized circuit cover sequences. We present an algorithm to get a generalized cutset (circuit) cover sequence from any given cutset (circuit) cover sequence. We then present GEN-CUTSET-SMART and GEN-CUTSET-SMART-SIMPLIFIED algorithms that remove some of the shortcomings of the dual framework of [21]. We prove that there is a one-to-one correspondence between the set of generalized circuit cover sequences and the set of generalized cutset cover sequences. We then show that for each execution of GEN-CIRCUIT-SMART there exists an execution of GEN-CUTSET-SMART-SIMPLIFIED such that the groups of edges that they map into edge disjoint lightpaths are exactly the same. In other words, the distinction between the primal and dual methods disappears when they use generalized sequences. Preliminary simulation results 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.

**Keywords:** Circuits, cutsets and duality; IP-over-WDM; lightpath routing; survivable networks; disjoint paths.

## I. INTRODUCTION

An IP-over-WDM network implements *Internet Protocol* (IP) directly over a *Wavelength Division Multiplexing* (WDM) network by mapping a set of given IP connections as *lightpaths* in the WDM network [1]. A lightpath is an all optical connection established by finding a path between the source and the destination of an IP connection in the WDM network and assigning it a wavelength [2]. Such networks use OXCs to switch network traffic (lightpaths) in the WDM layer and IP routers to route/reroute IP connections at the IP layer. The set of IP connections form the *logical topology* and OXCs along with actual optical fibers form the *physical topology*. In literature, it is common to refer to IP connections as *IP* or *logical links* (edges), IP routers as *logical nodes* (vertices), OXCs as *physical nodes* and set of fibers (a cable) connecting the OXCs as *physical links*. Fig. 1 shows a typical implementation of IP-over-WDM network.

An optical fiber simultaneously carries several lightpaths. Therefore, the failure of an optical fiber disconnects all the carried lightpaths, causing multiple

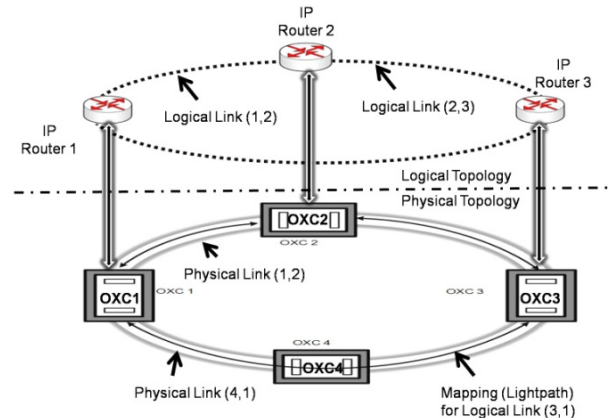


Figure 1. An IP-over-WDM Network.

The work of K. Thulasiraman has been supported in part by NSF grants ANI-0312435 and ECS 04-26831 and the work of G. Xue has been supported by NSF grants 0830739 and 0721803.

failures in the logical topology, which severely impacts the entire network performance. Mechanisms that allow networks to deliver an acceptable level of service in the presence of a failure or failures are referred to as *survivability mechanisms* and IP-over-WDM networks that implement such mechanisms are called *survivable IP-over-WDM networks* (henceforth, simply *survivable networks*). In this study, we consider link survivable networks that provide an acceptable level of service in the presence of a single physical link failure.

The two widely discussed survivability mechanisms in literature are *protection* and *restoration* [1]. Protection is generally provided at the physical layer but can be implemented at the logical layer also. It requires a dedicated *backup lightpath* for each *working lightpath* such that the two lightpaths are disjoint. The backup path is used only when the working lightpath fails. It is always possible to find two disjoint lightpaths, if the physical topology is at least 2-edge connected [3]. Restoration is usually provided at the logical layer by setting up working lightpaths for the IP connections and then provisioning the physical network with some additional (spare) capacity that is used by the IP routers to find backup lightpaths for the failed working lightpaths. However, backup paths can be guaranteed only if the IP topology is initially embedded in such a way that it stays connected after a failure. This leads to the study of the survivable logical topology mapping problem.

The rest of the paper is organized as follows. In section II we present the survivable logical topology mapping (SLTM) problem and review the literature on this problem. In section III we review the unified algorithmic framework, based on cutsets and circuits, developed in our recent work. To address a shortcoming of the CUTSET-SMART algorithm of this work we introduce in section IV the concepts of generalized circuit cover and generalized cutset cover sequences. In section V we develop an algorithm to get a generalized cutset (circuit) cover sequence from any given cutset (circuit) cover sequence. We then present in section VI GEN-CUTSET-SMART and GEN-CUTSET-SMART-SIMPLIFIED algorithms that remove some of the

shortcomings of the CUTSET-SMART algorithm of [21]. In section VII we prove that there is a one to one correspondence between the set of generalized circuit cover sequences and the set of cutset cover sequences. Using this result, we show that the distinction between the primal and dual methods disappears when they both use generalized sequences. Simulations comparing the performance of this new algorithm with the previous algorithms in terms of the number of additional logical links to be added to guarantee survivability and execution times are presented in section VIII

## II. SLTM: SURVIVABLE LOGICAL TOPOLOGY MAPPING

The *Survivable Logical Topology Mapping* (SLTM) problem in an IP-over-WDM optical network is to map each link  $(u, v)$  in the logical topology (at the IP layer) into a lightpath between the nodes  $u$  and  $v$  in the physical topology (at the optical layer) such that failure of a physical link does not cause the logical topology to become disconnected. It is assumed that both the physical and logical topologies are 2-edge connected (in short, two-connected).

Fig. 2(a) and Fig. 2(b) show a logical topology and a physical topology, respectively. Fig. 2(c) shows an unsurvivable mapping of this logical topology. In this case, not all the mappings are disjoint and the logical topology is not survivable. For example, the failure of physical link  $(4, 5)$  disconnects the logical topology. Fig. 2(d) shows a survivable mapping. In this case also, it can be seen that not all the mappings are disjoint and a physical link failure may disconnect multiple logical links but the logical topology still remains connected. For example, if the physical link  $(5, 6)$  fails, logical links  $(2, 6)$  and  $(4, 6)$  get disconnected but it is possible to reach all the logical nodes through the remaining logical links. It can be observed that finding disjoint mappings for only the subset  $\{(1, 2), (2, 4), (4, 6), (6, 1)\}$  is sufficient to guarantee survivability in this example. The question then arises as to how to select the groups of logical links to be mapped into disjoint paths. The answer to this important question will be provided in III.

The above example illustrates the important role played by the pair-wise (mutually) disjoint paths problem in finding survivable mappings. The problem of finding pair-wise disjoint paths is well studied and is NP-complete in general [4]. However, it is possible to find pair-wise disjoint paths in some special cases e.g. when the topology is undirected three edge-connected and the number of pair-wise disjoint paths required is two [4].

In [5-6], Modiano and Narula-Tam formally show that the problem of finding survivable mappings is NP-complete for general as well as for ring logical topologies. Therefore, they provide *Integer Linear Programs* (ILPs) to find a solution. The ILP is based on the observation that a logical topology can get disconnected after the failure of a physical link only if the physical link carries an entire cut of the

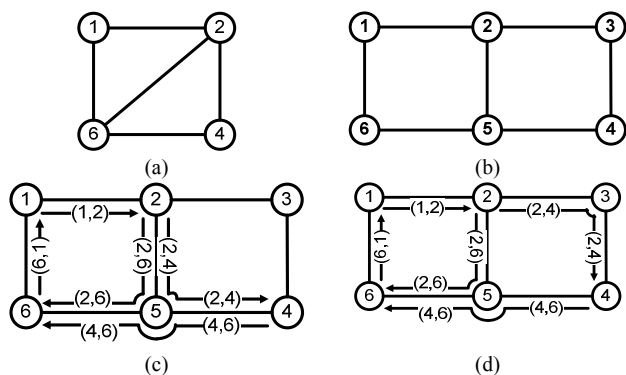


Figure 2. Illustration of mapping and survivability for general networks. (a) A logical topology (b) A physical topology (c) An unsurvivable mapping (d) A survivable mapping

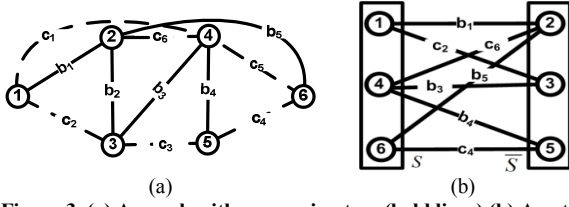


Figure 3. (a) A graph with a spanning tree (bold lines) (b) A cut.

logical topology, or alternatively, every cut of the logical topology must contain at least a pair of edges with pair-wise disjoint mappings in order for the mappings to be survivable. However, the ILP does not scale well as it must examine all the possible cuts, a number that grows exponentially with the size of the topology.

In [7] Todimala and Ramamurthy, based on [5-6], provide an improved ILP that applies to *Shared Risk Link Groups* (SRLG). The ILP incorporates wavelength assignment constraints and only considers primary cuts, but does not scale well either. However, when applied to planar cycles and hierarchical planar cycles, the ILP can be solved fairly quickly.

In [8] Ducatelle and Gambardella also utilize the results from [5-6] and rather than evaluating all the cutsets, employ a probability function as an estimate of the cutsets.

Crochat et al. provide in [9] a comprehensive framework for the logical topology mapping problem in IP-over-WDM networks and define three constraints (that include survivability) that must be respected by a solution. They note that the problem is NP-complete and suggest a heuristic based on Tabu search. Shenai and Sivalingam suggest in [10] a hybrid approach to survivability that uses a combination of restoration and protection.

In [11-13] Kurant and Thiran provide a framework called SMART (Survivable Mapping Algorithm by Ring Trimming). SMART utilizes circuits to find survivable mappings for logical topologies. The framework repeatedly picks connected *pieces* (subgraphs) of the logical topology and finds survivable mappings for these pieces. If a survivable mapping is found for a piece, its links are short-circuited (contracted) and the algorithm proceeds by picking another piece. The process is repeated until the logical topology is reduced to a single node or a search for a piece with survivable mapping is unsuccessful. In the former case, a survivable mapping for the logical topology has been found; otherwise a survivable mapping does not exist.

Some of the other works on the mapping problem may be found in [14-18].

### III. CIRCUITS/CUTSETS DUALITY AND A UNIFIED ALGORITHMIC FRAMEWORK FOR THE SLTM PROBLEM

Duality between circuits and cuts in a graph is one of the well-studied topics in graph theory. This concept has played a significant role in the development of methodologies for

solving problems in various applications. Most of the early results in electrical circuit theory were founded on the duality relationship between circuits and cuts [19]. There is a wealth of literature on the role of duality in network optimization (that is, discrete optimization on graphs and networks) [20]. Most often, for a primal algorithm based on circuits there is a dual algorithm based on cuts for the same problem. The primal and dual algorithms possess certain characteristics that make one superior to the other depending on the application. SMART algorithm for the survivable logical topology design problem is based on circuits [12]. The question then arises whether there exists a dual methodology based on cuts. The work in [21] answered this question in the affirmative and provided a unified algorithmic framework for the SLTM problem. The work also provided much insight into the structure of solutions for the SLTM problem. We discuss this framework in the following. We also present without proofs certain results from the graph theory literature that will be of interest in our developments in the following sections.

Given a spanning tree  $T$  with branches  $\{b_1, b_2, \dots, b_{n-1}\}$  and chords  $\{c_1, c_2, \dots, c_{m-n+1}\}$ , 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. With  $B(c_i)$  denoting the row corresponding to chord  $c_i$  the entry  $b_{ij}$  is defined as

$$b_{ij} = 1, \text{ if } B(c_i) \text{ contains edge } j, \\ = 0, \text{ otherwise.}$$

Arranging the rows of  $B_f$  such that the  $j^{\text{th}}$  row ( $j \leq m - n + 1$ ) corresponds to the fundamental circuit  $B(c_j)$  and arranging the columns in the order  $\{c_1, c_2, \dots, c_{m-n+1}, b_1, b_2, \dots, b_{n-1}\}$  we can write the  $B_f$  matrix as  $B_f = [U|B_{f_t}]$ , 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. 3(a) is given in (1).

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

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

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. With  $Q(b_i)$  denoting the row corresponding to branch  $b_i$  the entry  $q_{ij}$  is defined as

$$q_{ij} = 1, \text{ if } Q(b_i) \text{ contains edge } j \\ = 0, \text{ otherwise.}$$

Arranging the rows of  $Q_f$  such that the  $j^{\text{th}}$  row corresponds to  $f$ -cutset  $Q(b_j)$  and the columns correspond to edges in the order  $\{b_1, b_2, \dots, b_{n-1}, c_1, c_2, \dots, c_{m-n+1}\}$  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. 3(a) is given in (2).

In the following definitions,  $B(c_i)$  and  $Q(b_i)$  will also be used to denote the sets of edges in the corresponding fundamental circuit and fundamental cutset, respectively.

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

- a)  $[B(c_j) - c_j - \cup_{p=1}^{j-1} B(c_p)] \neq \emptyset, 2 \leq j \leq k$
- b)  $\cup_{p=1}^k B(c_p) = E - \{\text{chords not in the } B - \text{sequence}\}$

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), following are the three  $B$ -sequences:

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

Note that the order in which the  $B(c_j)$ 's appear matters in the definition of  $B$ -sequences. Without loss of generality assume that  $B(c_1), B(c_2), \dots, B(c_k)$  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)  $S(c_j) = B(c_j) - c_j - \cup_{p=1}^{j-1} B(c_p), 2 \leq j \leq k$

Then the submatrix of the  $f$ -circuit comprised of the rows corresponding to  $B(c_1), B(c_2), \dots, B(c_k)$  will have the structure shown in (3). Let the set of chords not in the circuit cover sequence be called **unmapped chords**.

An ordered sequence  $Q(b_1), Q(b_2), \dots, Q(b_k)$  is a **cutset cover sequence or simply a Q-sequence** of length  $k$  if

- a)  $[Q(b_j) - b_j - \cup_{p=1}^{j-1} Q(b_p)] \neq \emptyset, 2 \leq j \leq k$
- b)  $\cup_{p=1}^k Q(b_p) = E - \{\text{branches not in the } Q - \text{sequence}\}$

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), following are the three  $Q$ -sequences.

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

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

- a)  $\hat{S}(b_1) = Q(b_1) - b_1$
- b)  $\hat{S}(b_j) = Q(b_j) - b_j - \cup_{p=1}^{j-1} Q(b_p), 2 \leq j \leq k$

Then the submatrix of the  $f$ -cutset comprised of the rows corresponding to  $Q(b_1), Q(b_2), \dots, Q(b_k)$  has a structure similar to (3) as shown in (4). Let the set of branches not in the cutset cover sequence be called **unmapped branches**.

We now summarize a few standard results that will be of interest in the developments in the following sections.

### Theorem 1:

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

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

$$\begin{array}{cccccccccccc} c_1 & \dots & c_j & \dots & c_k & S(c_1) & S(c_2) & \dots & S(c_j) & S(c_k) \\ c_1 & [1 & 0 & \dots & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ c_2 & [0 & 1 & \dots & 0 & 0 & \times & \times & \times & \times & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ c_j & [0 & 0 & \dots & 1 & 0 & \times & \times & \times & \times & \times & \times & \times & \times & \times & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ c_k & [0 & 0 & 0 & 0 & \dots & 1 & \times & \times & \times & \times & \times & \times & \times & \times & \times & \times & \times & \times & 1 & 1 & 1 & 1 \end{array} \quad (3)$$

$$\begin{array}{cccccccccccc} b_1 & \dots & b_j & \dots & b_k & \hat{S}(b_1) & \hat{S}(b_2) & \dots & \hat{S}(b_j) & \hat{S}(b_k) \\ b_1 & [1 & 0 & \dots & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ b_2 & [0 & 1 & \dots & 0 & 0 & \times & \times & \times & \times & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ b_j & [0 & 0 & \dots & 1 & 0 & \times & \times & \times & \times & \times & \times & \times & \times & \times & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ b_k & [0 & 0 & 0 & 0 & \dots & 1 & \times & \times & \times & \times & \times & \times & \times & \times & \times & \times & \times & \times & 1 & 1 & 1 & 1 \end{array} \quad (4)$$

In the above  $\times$  means 0 or 1.

**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  $Q_{fc}$ . ■

**Theorem 4:** (a) Given a  $B$ -sequence  $B(c_1), B(c_2), \dots, B(c_k)$ , let  $B(c_{i_1}), B(c_{i_2}), \dots, B(c_{i_l})$  be a sub-sequence of this sequence then  $S(c_{i_1}) \subseteq B(c_{i_1}) \oplus B(c_{i_2}) \oplus \dots \oplus B(c_{i_l})$ .

(b) Given a  $Q$ -sequence  $Q(b_1), Q(b_2), \dots, Q(b_k)$ , let  $Q(b_{i_1}), Q(b_{i_2}), \dots, Q(b_{i_l})$  be a subsequence of this sequence then  $\hat{S}(b_{i_1}) \subseteq Q(b_{i_1}) \oplus Q(b_{i_2}) \oplus \dots \oplus Q(b_{i_l})$ . ■

**Theorem 5 [19]:** A graph is connected if and only if every cut of the graph contains at least one edge. ■

Using the above concepts several algorithms were proposed in [21] but only four algorithms CIRCUIT-SMART (Fig. 4), CUTSET-SMART (Fig. 5), CUTSET-SMART-SIMPLIFIED (Fig. 6) and INCIDENCE-SMART (Fig. 7) are discussed below. To guarantee survivability, these algorithms add to the logical graph new edges in parallel to some of the edges whenever necessary. These

edges will be called *protection edges*. The input to these algorithms is a physical (WDM) topology  $G_p$  and a logical (IP) topology  $G_L$ . The output of these algorithms is a survivable logical graph  $G'_L$  containing  $G_L$ .

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 step 1 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 step 1 does 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, if step 1 of this algorithm does 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. The rest of the paper addresses this question and provides an affirmative answer.

#### IV. GENERALIZED CIRCUIT/CUTSET COVER SEQUENCES

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

##### Algorithm CIRCUIT-SMART

###### 1. FOR $i = 1, 2, \dots, k$ DO

Map a maximum subset of edges in  $S(c_i) \cup c_i$  into disjoint lightpaths in  $G_p$  (see [22]). To all other edges in  $S(c_i) \cup c_i$  add protection edges and map each edge and its protection edge into disjoint lightpaths in  $G_p$ .

**END FOR**

2. Map all the chords not in the  $B$ -sequence into lightpaths in  $G$  arbitrarily.

**END**

Figure 4. Algorithm CIRCUIT-SMART

##### Algorithm CUTSET-SMART

###### 1. FOR $i = 1, 2, \dots, k$ DO

Map a maximum subset of edges in  $\hat{S}(b_i) \cup b_i$  into disjoint lightpaths in  $G_p$ . To all other edges in  $\hat{S}(b_i) \cup b_i$  add protection edges and map each edge and its protection edge into disjoint lightpaths in  $G_p$ . (see [22]).

**END FOR**

2. To each unmatched branch  $b$  add a protection edge  $b'$  and map them into disjoint lightpaths in  $G_p$ .

**END**

Figure 5. Algorithm CUTSET-SMART

##### Algorithm CUTSET-SMART-SIMPLIFIED

###### 1. FOR $i = 1, 2, \dots, k$ DO

Map  $b_i$  in disjoint manner with some chord in set  $\hat{S}(b_i)$ . If this is not possible for any chord in  $\hat{S}(b_i)$  then add a protection edge to one of these chords and map the chord and the protection edge in disjoint manner.

**END FOR**

2. To each unmatched branch  $b$  add a protection edge  $b'$  and map them as disjoint lightpaths in  $G_p$ .

3. Map all the unmapped logical edges arbitrarily.

**END**

Figure 6. Algorithm CUTSET-SMART-SIMPLIFIED

##### Algorithm INCIDENCE-SMART

###### 1. FOR $i = 1, 2, \dots, k$ DO

1. If vertex  $v_i$  has degree greater than or equal to 2 in the current graph, then map all the edges incident on it into disjoint lightpaths in  $G_p$ .

2. If the degree of  $v_i$  in the current graph is 1, then add a new logical edge connecting  $v_i$  to the datum vertex. Then map this new edge and the only edge incident on  $v_i$  into disjoint lightpaths.

3. If degree of  $v_i$  in the current graph is zero add two new parallel logical edges connecting  $v_i$  to the datum vertex. Then map these two edges into disjoint lightpaths in  $G_p$ .

**END FOR**

**END**

Figure 7. Algorithm INCIDENCE-SMART

- if this sequence is a circuit cover sequence, and
- for every unmapped chord  $c_i, B(c_i) \cap S(c_j) = S(c_j)$ , where  $j$  is the largest index such that  $B(c_i) \cap S(c_j) \neq \emptyset$ . 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  $Q(b_1), Q(b_2), \dots, Q(b_k)$  is a **generalized cutset cover sequence**

- if this sequence is a cutset cover sequence, and
- for every unmapped branch  $b_i, Q(b_i) \cap \hat{S}(b_j) = \hat{S}(b_j)$ , where  $j$  is the largest index such that  $Q(b_i) \cap \hat{S}(b_j) \neq \emptyset$ . 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  $B(c_1), B(c_2), \dots, B(c_k)$ . We define the set  $B - Cover(c_i)$ , for each  $i = 1, 2, \dots, k$  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  $B - Cover(c_1), B - Cover(c_2), \dots, B - Cover(c_k)$  in that order and arrange the columns to correspond to the sets  $B - Cover(c_1), B - Cover(c_2), \dots, B - Cover(c_k), S(c_1), S(c_2), \dots, S(c_k)$  then the  $f$ -circuit matrix will have the form shown in (5). In this figure, 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  $B - Cover(c_i)$ .

$$\begin{array}{cccccccccccccccc}
 B(c_1) & B(c_2) & B(c_3) & \dots & \dots & B(c_{i-1}) & B(c_i) & S(c_1) & S(c_2) & S(c_3) & \dots & \dots & S(c_{i-1}) & S(c_i) \\
 \hline
 U & 0 & 0 & \dots & \dots & 0 & 0 & I & \dots & 0 & \dots & 0 & \dots & 0 & 0 \\
 0 & U & 0 & \dots & \dots & 0 & 0 & \times & \dots & I & \dots & 0 & \dots & 0 & 0 \\
 0 & 0 & U & \dots & \dots & 0 & 0 & \times & \dots & \times & \dots & I & \dots & 0 & 0 \\
 \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
 \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
 0 & 0 & 0 & \dots & \dots & U & 0 & \times & \times & \times & \times & \times & \times & \times & \times & I & 0 \\
 0 & 0 & 0 & \dots & \dots & 0 & U & \times & \times & \times & \times & \times & \times & \times & \times & \times & I
 \end{array} \quad (5)$$

$$\begin{array}{cccccccccccccccc}
 Q(b_1) & Q(b_2) & Q(b_3) & \dots & \dots & Q(b_{i-1}) & Q(b_i) & \hat{S}(b_1) & \hat{S}(b_2) & \hat{S}(b_3) & \dots & \dots & \hat{S}(b_{i-1}) & \hat{S}(b_i) \\
 \hline
 U & 0 & 0 & \dots & \dots & 0 & 0 & I & \dots & 0 & \dots & 0 & \dots & 0 & 0 \\
 0 & U & 0 & \dots & \dots & 0 & 0 & \times & \dots & I & \dots & 0 & \dots & 0 & 0 \\
 0 & 0 & U & \dots & \dots & 0 & 0 & \times & \dots & \times & \dots & I & \dots & 0 & 0 \\
 \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
 \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
 0 & 0 & 0 & \dots & \dots & U & 0 & \times & \times & \times & \times & \times & \times & \times & \times & I & 0 \\
 0 & 0 & 0 & \dots & \dots & 0 & U & \times & \times & \times & \times & \times & \times & \times & \times & \times & I
 \end{array} \quad (6)$$

Similarly, given a generalized cutset cover sequence  $Q(b_1), Q(b_2), \dots, Q(b_k)$ , we define the set  $Q - Cover(b_i)$  for each  $i = 1, 2, \dots, k$  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  $Q - Cover(b_1), Q - Cover(b_2), \dots, Q - Cover(b_k)$  in that order and arrange the columns to correspond to the sets  $Q - Cover(b_1), Q - Cover(b_2), \dots, Q - Cover(b_k), \hat{S}(b_1), \hat{S}(b_2), \dots, \hat{S}(b_k)$ , then the  $f$ -cutset matrix will have the form shown in (6). In this figure, 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  $Q - Cover(b_i)$ .

#### V. GENERALIZED CIRCUIT/CUTSET COVER SEQUENCE ALGORITHM

Given a cutset cover sequence  $Q(b_1), Q(b_2), \dots, Q(b_k)$  of length  $k$  with respect to a spanning tree of a graph, we now present an algorithm 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,

$$\begin{aligned}
 GEN\_CUTSET\_SEQ(i) &= b_i, 1 \leq i \leq k \text{ and} \\
 GEN\_CUTSET\_SEQ(i) &= 0, k+1 \leq i \leq n-1.
 \end{aligned}$$

The set  $UNMAP$  will denote the set of branches that are not in the cutset cover sequence. So, initially  $UNMAP = \{b_{k+1}, b_{k+2}, \dots, b_{n-1}\}$ . 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  $Q(b_x) \cap \hat{S}(b_y) \neq \emptyset$ .

- If  $Q(b_x) \cap \hat{S}(b_y) \neq \hat{S}(b_y)$ , then the algorithm inserts  $b_x$  as the  $j^{th}$  element in  $GEN\_CUTSET\_SEQ$  and increments the positions of all subsequent elements by one. The algorithm also sets  $\hat{S}(b_x)$  to  $Q(b_x) \cap \hat{S}(b_y)$ . For example, if the current cutset cover sequence is  $b_1, b_2, \dots, b_{j-1}, b_y, b_{j+1}, \dots, b_k$  then the new sequence is  $b_1, b_2, \dots, b_{j-1}, b_x, b_y, b_{j+1}, \dots, b_k$ .
- Otherwise, that is, if  $Q(b_x) \cap \hat{S}(b_y) = \hat{S}(b_y)$ , then the algorithm adds  $b_x$  to  $Q - Cover(b_y)$ .

#### Theorem 6:

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. Step 1 in the algorithm guarantees 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 2 in the algorithm). At this point  $Q(b_x) \cap \hat{S}(b_y) = \hat{S}(b_y)$  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

#### Algorithm Gen-Cutset-Cover:

1. Pick a branch  $b_x \in UNMAP$ .  
Remove  $b_x$  from  $UNMAP$ .  
Let  $j$  be the largest index such that  $GEN\_CUTSET\_SEQ(j) = b_y$  and  $Q(b_x) \cap \hat{S}(b_y) \neq \emptyset$ .  
If  $Q(b_x) \cap \hat{S}(b_y) \neq \hat{S}(b_y)$ , then
    - $GEN\_CUTSET\_SEQ(j) \leftarrow b_x$ ,
    - $GEN\_CUTSET\_SEQ(r) \leftarrow GEN\_CUTSET\_SEQ(r-1), j+1 \leq r \leq n-1$ ,
    - $\hat{S}(b_x) \leftarrow Q(b_x) \cap \hat{S}(b_y)$ , and
    - $\hat{S}(b_y) \leftarrow \hat{S}(b_y) - \hat{S}(b_x)$
    - $Q - Cover(b_x) \leftarrow \{b_x\}$**Note:** The  $GEN\_CUTSET\_SEQ$  has been modified as  $[b_1, b_2, \dots, b_{j-1}, b_x, b_j, \dots, b_k, 0, 0, \dots, 0]$
  2. Otherwise, add  $b_x$  to  $Q - Cover(b_y)$
  3. Repeat steps 1 and 2 until  $UNMAP$  is empty
- END**

Figure 8. Algorithm Gen-Cutset-Cover

**Algorithm GEN-CUTSET-SMART:****Step 0:**

Starting with any cutset cover sequence generate a generalized cutset cover sequence using algorithm Gen-Cutset-Cover. Let this sequence be  $Q(b_1), Q(b_2), \dots, Q(b_k)$ .

**For each**  $b_i, i = 1, 2, \dots, k$

Map a maximum subset of edges in  $Q - \text{Cover}(b_i) \cup \hat{S}(b_i)$  into disjoint lightpaths in  $G_p$ . To all other edges in  $Q - \text{Cover}(b_i) \cup \hat{S}(b_i)$  add protection edges and map each edge and its protection edge into disjoint lightpaths in  $G_p$ . (see [22]).

**END**

Figure 9. Algorithm GEN-CUTSET-SMART

corresponding  $\hat{S}(b_y)$  with  $Q(b_x)$  is  $\hat{S}(b_y)$ . So  $b_x$  continues to be in  $Q - \text{Cover}(b_y)$ . Thus, at the end of the algorithm every branch is in some  $Q - \text{Cover}(b_i)$  and no branch is in more than one  $Q - \text{Cover}$  set guaranteeing that the algorithm produces a generalized cutset cover sequence. ■

## VI. 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 2 of CUTSET-SMART and has a modified version of step 1 of CUTSET-SMART.

We now prove the correctness of algorithm GEN-CUTSET-SMART. In Theorem 7,  $G'_L$  refers to the original logical topology  $G_L$  augmented with protection edges added as in Step 1 of algorithm GEN-CUTSET-SMART.

**Theorem 7:** 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

**Algorithm GEN-CUTSET-SMART-SIMPLIFIED****Step 0:**

Starting with any cutset cover sequence generate a generalized cutset cover sequence using algorithm Gen-Cutset-Cover. Let this sequence be  $Q(b_1), Q(b_2), \dots, Q(b_k)$ .

**Step 1:**

**For each**  $b_i, i = 1, 2, \dots, k$

Pick a chord  $c_r$  in set  $\hat{S}(b_i)$ . Map a maximum subset of edges in  $Q - \text{Cover}(b_i) \cup c_r$  into edge disjoint lightpaths in  $G_p$ . To all other edges in  $Q - \text{Cover}(b_i) \cup c_r$  add protection edges and map each edge and its protection edge into disjoint lightpaths in  $G_p$ . (see [22]).

Figure 10. Algorithm GEN-CUTSET-SMART-SIMPLIFIED

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 5.

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  $b_k \in Q - \text{Cover}(b_i)$  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 5.

If the cut has no branch  $b_k \in Q - \text{Cover}(b_i)$  with  $i \neq k$ , then by Theorem 4(b) the set  $\hat{S}(b_j)$  will be in the cut. Since the branch  $b_j$  and the edges in the set  $\hat{S}(b_j)$  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 5.

Thus in all cases, the algorithm generates a survivable mapping of the graph  $G'_L$ . ■

A generalized version of CUTSET-SMART-SIMPLIFIED similar to GEN-CUTSET-SMART can be designed and is given in Fig. 10. 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.

## VII. 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  $Q(b_1), Q(b_2), \dots, Q(b_k)$ . Recall that  $Q - \text{Cover}(b_i)$  refers to the set of branches covered by the branch  $b_i$  in the cover sequence. Note that all branches in  $Q - \text{Cover}(b_i)$  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  $Q - \text{Cover}(b_1), Q - \text{Cover}(b_2), \dots, Q - \text{Cover}(b_k)$  in that order and arrange the columns to correspond to the sets  $Q - \text{Cover}(b_1), Q - \text{Cover}(b_2), \dots, Q - \text{Cover}(b_k), \hat{S}(b_1), \hat{S}(b_2), \dots, \hat{S}(b_k)$ , then the submatrix  $Q_{f_c}^t$  of  $Q_f$  whose columns are arranged in the order  $\hat{S}(b_1), \hat{S}(b_2), \dots, \hat{S}(b_k)$ , will have the form shown in (7) (See also equation (6)).

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

Consider now the fundamental circuit matrix  $B_f$ . Then the submatrix  $B_{ft}$  with its columns arranged to correspond to the sets  $Q\text{-Cover}(b_k), Q\text{-Cover}(b_{k-1}), \dots, Q\text{-Cover}(b_1)$  in that order and the rows arranged to correspond to the sets  $\hat{S}(b_k), \hat{S}(b_{k-1}), \dots, \hat{S}(b_1)$ , 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 (7). This is shown in (8), where for the sake of convenience,  $\hat{Q}(b_i)$  stands for  $Q\text{-Cover}(b_i)$ . This leads to Theorem 8.

$$Q_{fc} = \begin{matrix} & \hat{S}(b_1) & \hat{S}(b_2) & \dots & \hat{S}(b_{k-1}) & \hat{S}(b_k) \\ \begin{matrix} I & 0 & \dots & 0 & 0 \\ X & I & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ X & X & \dots & I & 0 \\ X & X & \dots & X & I \end{matrix} & \dots & (7) \end{matrix}$$

**Theorem 8:** Given a generalized cutset cover sequence  $Q(b_1), Q(b_2), \dots, Q(b_k)$ , let chords  $c_1, c_2, \dots, c_k$  be selected such that each  $c_i \in \hat{S}(b_j)$ , where  $j = k-i+1$ . Then the sequence  $B(c_1), B(c_2), \dots, B(c_k)$  is a generalized circuit cover sequence with

- $B\text{-Cover}(c_i) = \hat{S}(b_{k-i+1})$ ; and
- $S(c_i) = Q\text{-Cover}(b_{k-i+1})$ . ■

$$B_{ft} = \begin{matrix} & \hat{Q}(b_k) & \hat{Q}(b_{k-1}) & \dots & \hat{Q}(b_2) & \hat{Q}(b_1) \\ \begin{matrix} I & 0 & \dots & 0 & 0 \\ X & I & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ X & X & \dots & I & 0 \\ X & X & \dots & X & I \end{matrix} & \dots & (8) \end{matrix}$$

Starting from a generalized circuit cover sequence one can also get a generalized cutset cover sequence and a result dual to Theorem 8 is stated in Theorem 9.

**Theorem 9:** Given a generalized circuit cover  $B(c_1), B(c_2), \dots, B(c_k)$ , let branches  $b_1, b_2, \dots, b_k$  be selected such that each  $b_i \in S(c_{k-i+1})$ . Then

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

In view of Theorems 8 and 9, we get the following.

**Theorem 10 (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 step 1 by algorithm CIRCUIT-SMART corresponds to the set of edges picked (for disjoint mappings) in an execution of step 1 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

$$\begin{aligned} B\text{-Cover}(c_2) &= \{c_2\} & , & S(c_2) = \{b_1, b_2\} \\ B\text{-Cover}(c_6) &= \{c_6, c_1\} & , & S(c_6) = \{b_3\} \\ B\text{-Cover}(c_3) &= \{c_3\} & , & S(c_3) = \{b_4\} \\ B\text{-Cover}(c_4) &= \{c_4, c_5\} & , & S(c_4) = \{b_5\} \end{aligned}$$

Picking the branches selected as in Theorem 9, we can get the following generalized cutset cover sequence  $Q(b_5), Q(b_4), Q(b_3), Q(b_2)$  with

$$\begin{aligned} Q\text{-Cover}(b_5) &= \{b_5\} & , & \hat{S}(b_5) = \{c_4, c_5\} \\ Q\text{-Cover}(b_4) &= \{b_4\} & , & \hat{S}(b_4) = \{c_3\} \\ Q\text{-Cover}(b_3) &= \{b_3\} & , & \hat{S}(b_3) = \{c_1, c_6\} \\ Q\text{-Cover}(b_2) &= \{b_1, b_2\} & , & \hat{S}(b_2) = \{c_2\} \end{aligned}$$

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  $\hat{S}(b_5)$  and  $c_6$  from  $\hat{S}(b_3)$  then we will select the above sets of edges for disjoint mappings as stated in Theorem 10.

In view of Theorem 10 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 step 1 of GEN-CUTSET-SMART-SIMPLIFIED by the following:

**a) Dual Step (Cutset based):**

if  $Q\text{-Cover}(b_i)$  has exactly one branch, namely, itself, then pick any chord  $c_i$  in  $\hat{S}(b_i)$  and map  $c_j$  and  $b_i$  into disjoint lightpaths in the physical topology.

**b) Primal Step (Circuit based):**

if  $Q\text{-cover}(b_i)$  has more than one branch, then pick any chord  $c_j$  in  $\hat{S}(b_i)$ , and map  $c_j$  and all the branches in  $Q\text{-cover}(b_i)$  into disjoint light paths in the physical topology.

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

## VIII. SIMULATIONS AND RESULTS

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 [19]. 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 [22] was used. To find mutually disjoint mappings of a pair of logical links, the algorithm given in [4] was used. Fundamental circuits and cutsets were found using procedures given in [19] 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 I and Table II. 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

Logical Degree	Algorithm	Protection Edges	Execution Time (sec)
2.5	CIRCUIT-SMART	15.99	0.54344
	CUTSET-SMART-SIMPLIFIED	26.23	0.07191
	GEN-CUTSET-SMART-SIMPLIFIED	12.93	0.50866
3.0	CIRCUIT-SMART	8.86	0.39454
	CUTSET-SMART-SIMPLIFIED	20.86	0.08565
	GEN-CUTSET-SMART-SIMPLIFIED	7.48	0.40393
3.5	CIRCUIT-SMART	5.98	0.3275
	CUTSET-SMART-SIMPLIFIED	16.1	0.09735
	GEN-CUTSET-SMART-SIMPLIFIED	5.5	0.31249
4.0	CIRCUIT-SMART	3.92	0.2594
	CUTSET-SMART-SIMPLIFIED	12.66	0.10701
	GEN-CUTSET-SMART-SIMPLIFIED	4.74	0.28124

Logical Degree	Algorithm	Protection Edges	Execution Time (sec)
2.5	CIRCUIT-SMART	41.59	4.92908
	CUTSET-SMART-SIMPLIFIED	56.57	0.30911
	GEN-CUTSET-SMART-SIMPLIFIED	35.8	4.76246
3.0	CIRCUIT-SMART	25.14	3.62206
	CUTSET-SMART-SIMPLIFIED	44.04	0.39299
	GEN-CUTSET-SMART-SIMPLIFIED	20.4	3.41603
3.5	CIRCUIT-SMART	14.07	2.50407
	CUTSET-SMART-SIMPLIFIED	36.08	0.46103
	GEN-CUTSET-SMART-SIMPLIFIED	11.48	2.4402
4.0	CIRCUIT-SMART	10.67	2.25127
	CUTSET-SMART-SIMPLIFIED	26.72	0.52782
	GEN-CUTSET-SMART-SIMPLIFIED	9.44	2.15848

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.

## IX. SUMMARY

In this paper we have carried out a theoretical investigation of the survivable logical topology mapping (SLTM) problem. In [12] Kurant and Thiran had provided an algorithmic framework called SMART for this problem. SMART requires successive contraction of circuits in the logical graph and mapping the links in these circuits into edge disjoint lightpaths in the physical topology. In a recent work we developed a dual framework based on cutsets and showed that both the primal framework (SMART) based on circuits and the dual framework based on cutsets have the same algorithmic structure. We also presented algorithms CIRCUIT-SMART, CUTSET-SMART, CUTSET-SMART-SIMPLIFIED and INCIDENCE-SMART. The effectiveness of both these frameworks in terms of the additional logical edges (called protection edges) to be added to guarantee survivability of the logical graph, and their robustness in terms of their ability to survive multiple physical link failures depend on the lengths of the circuit cover and cutset cover sequences on which they are based. In this paper we studied this issue. First, we introduced the concepts of generalized circuit cover and generalized cutset cover sequences. We presented an algorithm to generate a generalized cutset (circuit) cover sequence from an arbitrary cutset (circuit) cover sequence. We then presented generalized versions of CUTSET-SMART, CUTSET-SMART-SIMPLIFIED called GEN-CUTSET-SMART and GEN-CUTSET-SMART-SIMPLIFIED, respectively. Algorithm CIRCUIT-SMART remains the same whether we use a circuit cover sequence or a generalized circuit cover

sequence. We showed using the duality relationship between circuits and cutsets that each generalized circuit cover sequence corresponds to a generalized cutset cover sequence. We then showed that for each execution of CIRCUIT-SMART there exists an execution of GEN-CUTSET-SMART-SIMPLIFIED such that the groups of edges they map into edge disjoint lightpaths are exactly the same. In other words, the distinction between the primal and dual methods disappears when they both use generalized sequences. This means that GEN-CUTSET-SMART-SIMPLIFIED will have better performance 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. Our preliminary simulations confirm this. Research is in progress on the impact of these generalizations on the robustness of these schemes in terms of their ability to survive multiple physical link failures.

#### REFERENCES

- [1] L.Sahasrabudde, S. Ramamurthy and B. Mukherjee, "Fault Management in IP-Over-WDM Networks: WDM Protection versus IP Restoration", *IEEE Journal on Selected Areas in Communications*, Vol. 20, No.1, pp. 21-33, 2002.
- [2] I. Chlamtac, A. Ganz and G. Karmi, "Lightpath Communications: An Approach to High Bandwidth Optical WDM", *IEEE Transactions on Communications*, Vol. 40, No.7, pp. 1171-1182, 1992.
- [3] D. West, *Introduction to Graph Theory*, Prentice Hall, 1996.
- [4] Y. Perl and Y. Shiloach, "Finding two disjoint paths between two pairs of vertices in a graph", *Journal of the ACM*, Vol. 25, No.1, pp. 1-9, Jan. 1978.
- [5] E. Modiano and A. Narula-Tam, "Survivable Lightpath Routing: A New Approach to the Design of WDM-based Networks", *IEEE Journal on Selected Areas in Communications*, Vol. 20, No. 4, pp. 800-809, 2002.
- [6] E. Modiano and A. Narula-Tam, "Survivable Routing of Logical Topologies in WDM Networks", *IEEE INFOCOM 2001*, pp. 348-357.
- [7] A. Todimala and B. Ramamurthy, "A Scalable Approach for Survivable Virtual Topology Routing in Optical WDM Networks", *IEEE Journal on Selected Areas in Communications*, Vol. 25, No. 6, pp. 63-69, 2007.
- [8] F. Ducatelle and L. Gambardella, "FastSurv: A New Efficient Local Search Algorithm for Survivable Routing in WDM Networks", *IEEE Globecom 2004*, pp. 1925-1929.
- [9] O. Crochat, J. Boudec and O. Gerstel, "Protection Interoperability for WDM Optical Networks", *IEEE/ACM Transactions on Networking*, Vol. 8, No.3, pp. 384-395, 2000.
- [10] R. Shenai and K. Sivalingam, "Hybrid Survivability Approaches for Optical WDM Mesh Networks", *Journal of Lightwave Technology*, Vol. 23, No. 10, pp. 3046-3055, 2005.
- [11] M. Kurant and P. Thiran, "Survivable Mapping Algorithm by Ring Trimming (SMART) for Large IP-over-WDM Networks", *Broadband Networks 2004*, pp. 44-53.
- [12] M. Kurant and P. Thiran, "On Survivable Routing of Mesh Topologies in IP-over-WDM Networks", *IEEE INFOCOM 2005*, pp. 1106-1116.
- [13] M. Kurant and P. Thiran, "Survivable Routing of Mesh Topologies in IP-over-WDM Networks by Recursive Graph Contraction", *IEEE Journal on Selected Areas in Communications*, Vol. 25, No. 5, pp. 922-933, 2007.
- [14] C. Liu and L. Ruan, "A New Survivable Mapping Problem in IP-over-WDM Networks", *IEEE Journal on Selected Areas in Communications*, Vol. 25, No.3, pp. 25-34, 2007.
- [15] L. Ruan and F. Tang, "Survivable IP Network Realization in IP-over-WDM Networks under Overlay Model", *Computer Communications*, Vol. 29, No. 10, pp. 1772-1779, 2006.
- [16] C. Liu and L. Ruan, "Logical Topology Augmentation for Survivable Mapping in IP-over-WDM Networks", *IEEE Globecom 2005*, pp. 1885-1889.
- [17] L. Ruan and C. Liu, "A Heuristic Algorithm for Survivable Mapping in IP-over-WDM Networks", *Communications and Computer Networks*, pp. 164-170, 2005.
- [18] K. Lee and E. Modiano, "Cross-layer Survivability in WDM-based Networks", *IEEE Infocom 2009*, pp. 1017-1025.
- [19] K. Thulasiraman and M. N. S. Swamy, *Graphs: Theory and Algorithms*, Wiley, 1992.
- [20] R. Ahuja, T. Magnanti and J. Orlin, *Network Flows: Theory and Algorithms*, Prentice Hall, 1993.
- [21] K. Thulasiraman, M. Javed and G. Xue, "Circuits/Cutsets Duality and a Unified Algorithmic Framework for Survivable Logical Topology Design in IP-over-WDM Optical Networks", *INFOCOM 2009*, pp. 1026-1034.
- [22] J. Kleinberg, "Approximation Algorithms for Disjoint Paths Problems," MIT, Cambridge, MA, PhD Thesis, 1996.