Survivability of Lightwave Networks - Path Lengths in WDM Protection Scheme

Arunabha Sen*, Bin Hao and Bao Hong Shen
Department of Computer Science and Engineering
Arizona State University
Tempe, AZ 85287, USA

Subir Bandyopadhyay
School of Computer Science
University of Windsor
Windsor, Ontario, Canada

Abstract

In the *protection* scheme of fault management in a WDM optical network, corresponding to every source-destination path used for data transmission, a *backup path* is maintained in a stand-by mode. In the case of a failure (either due to a fiber cut or due to equipment failure) in the primary path, data transmission is quickly switched to the backup path. In order to tolerate any *single* fault, the backup path must be *edge (or node) disjoint* from the primary path. Most often a *shortest path* between the source and the destination is chosen as the *primary* path. To obtain a link (node) disjoint backup (or *secondary*) path, the links (nodes) of the primary path are removed from the graph and then a shortest path in the *modified* graph is chosen as the backup path. The attractive feature of this scheme is its simplicity. However, the scheme has a severe drawback. Due to the choice of a shortest path as the primary path, the length of a link disjoint secondary path may be unacceptably large. In this paper, we propose a novel way of choosing the primary and the secondary paths so that the lengths of both the paths are small. Unfortunately, the problem of choosing primary and secondary paths in this way turns out to be NP-complete. We provide the NP-completeness proof of both the edge disjoint and the node disjoint version of the problem. We provide an *approximation algorithm* for the problem with a guaranteed performance bound of 2 and a mathematical programming formulation for the *exact solution* of the problem. Though the approximate solution provides a performance bound of 2, through extensive experimental evaluation, we find that the approximate solution is very close to the optimal solution and the ratio between the approximate to the optimal solution never exceeds 1.2. Although we discuss the *single fault* scenario in this paper, the algorithms discussed here, can be used equally effectively for the *multiple fault* scenario also. Finally, we discuss other variations of the disjoint path problem relevant to the lightwave networks.

*Corresponding author: Arunabha Sen, Telephone: 480-965-6153, Fax: 480-965-2751, e-mail: asen@asu.edu

Keywords: Optical networks, WDM, survivability, min-max path pair, min-sum path pair, shortest path pair

1 Introduction

Survivability of high bandwidth optical networks has become an area of concern because of its importance as a national and international infrastructure for moving large volumes of data from one part of the globe to another. Failure of any part in such an important infrastructure and the resulting inability to move data around quickly may have a tremendous economic impact. For this reason, survivability issues in high bandwidth optical networks has become an important area of research in recent years. The research on on this topic has mostly focussed on SONET networks so far. Only recently, researchers have started examining survivability issues in WDM networks [2, 4, 7, 8, 18, 22]. Two techniques, *protection* at the WDM layer and *restoration* at the IP layer, have emerged as the leading contenders for fault management in optical networks [23]. Protection refers to pre-provisioned failure recovery (usually hardware based), whereas restoration refers to more dynamic recovery (usually software based) [12, 11]. Between the two schemes, protection is typically faster and usually performs single link protection.

In optical fiber networks, cuts in fibers are considered to be one of the most common failures. Since in a WDM network, an optical fiber cable carries an extremely high volume of traffic, disruption of service is very expensive. Equipment failure at a switching node disables all the links passing through that node and the adverse impact of such an event on the network is even greater than that of a cable cut. Due to its importance as a critical national (and international) infrastructure, quick detection and identification of

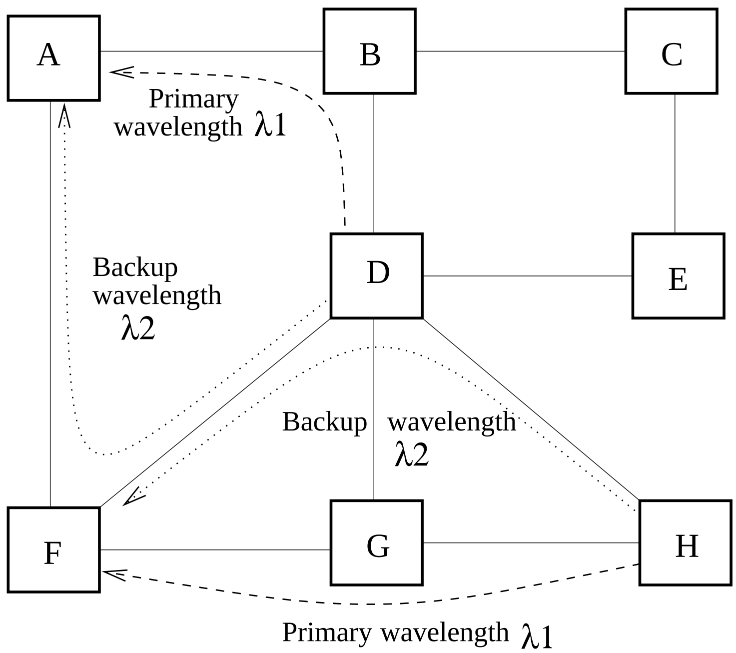

Figure 1: Primary and backup paths in a WDM network

faults and restoration of normal services is absolutely essential.

In the WDM protection scheme of fault management in an optical network, corresponding to every source-destination path used for data transmission, a *backup path* is maintained in a stand-by mode. In the *mesh path protection* scheme, global mesh protection is provided by switching, in less than 100ms, from a failed lightpath to a backup path. Due to very fast switching time constraint, these schemes are based on predefined backup paths designed ahead of time in a central location and stored in the nodes of the network, so that they can be activated quickly in the case of a failure of the primary path [10]. The primary and the backup paths between the nodes D to A and from H to F is shown in figure 1.

Clearly, in order to tolerate any single link failure, the alternate path should not be sharing any fiber link with the primary path. In order to tolerate any single node equipment failure, the alternate path should not be sharing any node with the primary path. Thus the alternate path should be *edge (or node) disjoint* from the primary path.

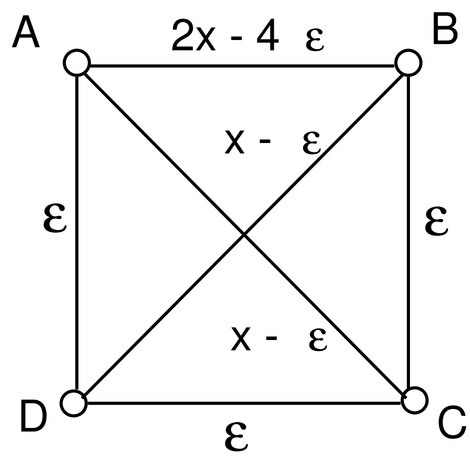

Figure 2: MinMax path and MinSum path in a WDM network

Since there are a number of virtues associated with a short path length, often a *shortest path* between the source and the destination is used as the *primary* path. To obtain a link (node) disjoint backup (or *secondary*) path, the links (nodes) of the primary path are removed from the graph and then a shortest source-destination path in the *modified* graph is used as the backup path. We will refer to this path pair as *Shortest Path Pair (SPP)*. Although this idea is attractive because of its simplicity, the selection of the primary and the secondary paths in this manner has a severe drawback. Because the primary path is a shortest path in the original graph, there is no control over the length of the secondary path and this length may be unacceptably large. If the path lengths represent the delay, this implies unacceptably long delay on the backup path. In the example shown in figure 2, node A is the source and node B is the destination. Suppose that the weights on the links shown in the figure represent the delay. In this example, the shortest path from A to B is $A \rightarrow D \rightarrow C \rightarrow B$ with a total delay of 3ϵ, where ϵ is a very small number. If this path is used as the primary path, the edge disjoint backup path has to be the direct link between A and B with a delay of $2x - 4\epsilon$, where x is any arbitrary number. However, in this example if the path $A \rightarrow C \rightarrow B$ is used as the primary path, then the path $A \rightarrow D \rightarrow B$ can be used as the backup path, with delays on both the paths being equal to x. It may be noted that the delay associated with the backup path in the first pair of disjoint paths $(2x - 4\epsilon)$, is almost double of the delay of the backup path on the second pair of paths (x). In a QoS constrained environment, it is extremely important that the data arrives at the destination within a specified delay. If the acceptable delay is x and there is a non-zero probability of failure of the primary path, then clearly the second pair of paths is much preferable than the first pair.

If all edge weights are identical, then the length of the paths is measured in terms of the number of *hops* between the source and the destination. Long paths are undesirable because it increases the blocking probability. The results presented in this paper are valid for both the situations, when (i) the path length is measured in terms of the delay and (ii) the path lengths are measured in terms of hops.

In this paper, we propose a novel way of choosing the primary and the backup paths such that the length of the primary as well as the backup path is small. Disjoint path problems, because of their applications in the design of reliable networks, have been extensively studied [3, 5, 19, 28, 30, 31]. The knowledge of a set of k disjoint (either node or edge) paths between a pair of source-destination nodes can be utilized to enhance the reliability and congestion control aspects of the network. Most of the earlier studies on this topic find k node or edge disjoint paths between a source-destination node pair such that the *sum of the path lengths* is minimized. In the WDM protection scheme we are interested in only two disjoint (primary and backup) paths, i.e., $k = 2$. We refer to the pair of disjoint paths whose sum of the path lengths is minimum as the *Minimum Sum Path Pair,(MinSumPP)*.

If the path lengths represent the *delay* on that path, minimizing the sum of the path lengths may not be very meaningful. In this case, one would like to find a set of k (node or edge)-disjoint paths between the source-destination node pair such that the length of the *longest* path in this set is *shortest* among all such sets. In the optical domain, where $k = 2$, we want to find the path pair, whose longer path is shortest among all such pairs of paths. We refer to this path pair as the *Minimum Maximum Path Pair,(MinMaxPP)*. In the WDM protection scheme, if MinMaxPP is selected as the primary and the backup paths, then the delay on the backup path will be minimum (the delay on the primary path is always less than that of the backup path).

In this paper we propose the use of MinMaxPP as the path pair for the primary and the backup path instead of the shortest path pair, SPP. Unfortunately, the problem of choosing a primary and a backup path in this way for a specified source-destination node pair turns out to be NP-complete. We provide the NP-completeness proof of both the edge disjoint and the node disjoint version of the MinMaxPP problem. In contrast, MinSumPP can be computed very efficiently by Dijkstra's shortest path computation algorithm [30, 31]. In fact, the computational complexity of the MinSumPP is no different from that of the SPP, whose main attraction is its simplicity. Since MinMaxPP is computationally difficult to find, we propose the use of MinSumPP as an *approximation* of the MinMaxPP. We show that the approximate solution (MinSumPP) never exceeds the optimal solution (MinMaxPP) by factor of 2.

In this paper we also provide a mathematical programming formulation for the *exact solution* of the MinMaxPP problem. Though the approximate solution (MinSumPP) provides a performance bound of 2, through extensive experimental evaluation, we find that the ratio between the approximate solution to the optimal solution never exceeds 1.2. Although we discuss *single fault* ($k = 2$) scenario in this paper, the algorithms discussed here, can be used equally effectively for the *multiple fault* scenario also (i.e., any arbitrary k). Finally, we discuss other variations of the disjoint path problem applicable to the lightwave networks.

The paper is organized as follows: in section 2 we describe the edge- and node-disjoint versions of the path problem and show that both the problems are NP-complete; in section 3 we provide a mathematical programming formulation for the *exact* solution of the problem; in section 4 we present a heuristic solution and show that the *approximate* solution obtained by this method is bounded by two times the optimal solution; section 5 presents the comparison results of our experimental evaluation of the performance of the exact solution with the approximate solution; in section 6 we discuss some variation of the backup path problem; section 7 concludes the paper.

2 Disjoint Path Problem

As mentioned earlier, disjoint path problems, because of their applications in the design of reliable networks, have been extensively studied [3, 5, 19, 28, 30, 31]. An algorithm for computing a set of k *edge-disjoint* paths between a pair of source and destination nodes, with a minimum total edge cost is given in [15]. Suurballe in [30] presented an algorithm for the same problem for *node-disjoint* paths. An efficient implementation of the edge-disjoint path problem with minimum total edge cost is given in [31]. Skiscim *et. al.* in [28] studied the problem of k successively shortest paths problem. However, in this case the paths are not required to be either node or edge disjoint. Nikolopoulos in [19] provided a solution for the selection of k-best (paths with minimum total edge cost) node disjoint paths by mapping it into a Trellis graph.

Suppose that the number of edge (node) disjoint paths between any pair of nodes

u and v in a graph $G = (V, E)$ is denoted by $\kappa(u, v)$. The *edge (node) connectivity* of a graph is the minimum value of $\kappa(u, v)$ taken over all pairs of nodes u and v. For an optical network to have a backup path corresponding to every source destination node pair, the connectivity of the network must be at least two. Topology design of optical networks has been extensively researched. Most of the topologies proposed have edge (node) connectivity of at least two [6, 13, 16, 17, 21, 24, 25, 26, 27, 32].

2.1 Node Disjoint Path Problem

Instance: A graph $G = (V, E)$, with weight $w_{i,j}$ associated with each edge $e_{i,j}$, ($w_{i,j} \geq 0$), a source node s, a destination node d and a positive integer k.

Problem: Find a set of k node-disjoint paths from s to d such that the length of the *longest* path in this set, is the *shortest* among all such sets of k node-disjoint paths from s to d.

Node Disjoint k-Path Problem (NDkPP) - *Decision Version*

Instance: A graph $G = (V, E)$, with weight $w_{i,j}$ associated with each edge $e_{i,j}$, ($w_{i,j} \geq 0$), a source node s, a destination node d and a positive integer k and a non-negative number X.

Question: Is there a set of k node-disjoint paths from s to d such that the length of the *longest* path in this set is less than or equal to X ?

Theorem 1 *The node disjoint k-path problem is* NP-complete.

Proof: We show NDkPP is NP-complete by considering a restricted version of the problem where $k = 2$. We call the restricted version the *Node Disjoint* 2-*Path Problem (ND2PP)*.

Node Disjoint 2-Path Problem (ND2PP)

Instance: A graph $G = (V, E)$, with weight $w_{i,j}$ associated with each edge $e_{i,j}$, $(w_{i,j} \geq 0)$, a source node s, a destination d and a non-negative number X.

Question: Is there a set of two node-disjoint paths from s to d such that the length of the *longest* path in this set is less than or equal to X ?

We show ND2PP is NP-complete by giving a transformation from a well known NP-complete problem - the *Partition* Problem [9].

Partition Problem (PartProb)

Instance: A finite set A and a "size" $s(a_i) \in Z^+$ for each $a_i \in A$.

Question: Is there a subset $A' \subseteq A$ such that

$$\sum_{a_i \in A'} s(a_i) = \sum_{a_i \in A - A'} s(a_i)$$

Proof: It is easy to see that ND2PP $\in$ NP, since a non-deterministic algorithm need only guess two edge disjoint paths P_1 and P_2 and check in polynomial time that the length of the longer path is less than or equal to X.

We will transform the Partition Problem (PartProb) to ND2PP. Let the set $A = \{a_1, \ldots, a_n\}$ and a "size" $s(a_i) \in Z^+$ for each $a_i \in A, 1 \leq i \leq n$, make up an arbitrary instance of the PartProb. From this instance of the PartProb we construct an instance of the ND2PP problem. An instance of the ND2PP comprises of a non-negative integer X and a graph $G = (V, E)$ with the following charactersitcs:

(i) each edge (i, j) has a non-negative weights $w_{(i,j)}$ associated with it,

(ii) the graph has specified source and destination nodes s and t respectively,

(iii) the graph $G = (V, E)$ has two node disjoint paths from s to t of length at most X, if and only if the elements of the set A can be partitioned into two groups such that the sum of the weights in one group is equal to the sum of the weights in the other group.

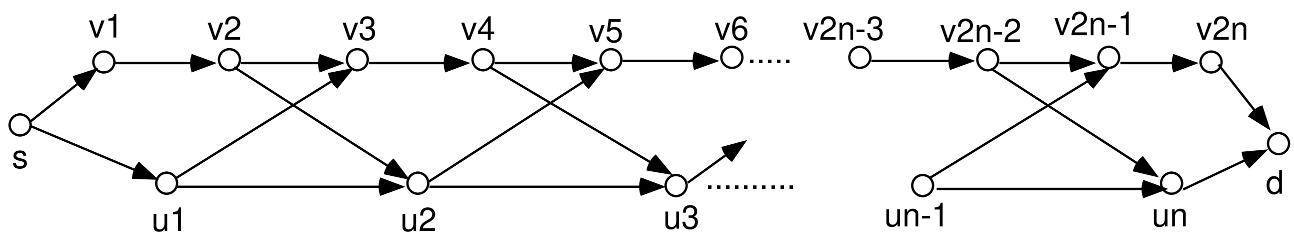

Figure 3: Node Disjoint Paths

We construct the graph $G = (V, E)$ in such a way that it has $3n + 2$ nodes, $V = \{\{v_i, 1 \leq i \leq 2n\} \cup \{u_i, 1 \leq i \leq n\} \cup \{s\} \cup \{d\}\}$. The directed edges are from

(i) $s = u_0 = v_0 \rightarrow v_1$ and $s \rightarrow u_1$

(ii) $v_{2n} \rightarrow d$ and $u_n \rightarrow d = u_{n+1} = v_{2n+1}$

(iii) $v_i \rightarrow v_{i+1}, 1 \leq i \leq 2n - 1$

(iv) $u_i \rightarrow u_{i+1}, 1 \leq i \leq n - 1$

(v) $v_{2i} \rightarrow u_{i+1}, 1 \leq i \leq n - 1$

(vi) $u_i \rightarrow v_{2i+1}, 1 \leq i \leq n - 1$

Such a graph is shown in figure 3. The weights on the edges are as follows:

(i) $w(v_{2i-1} \rightarrow v_{2i}) = s(a_i),\ 1 \leq i \leq n$

(ii) The weight of every other edge is 0.

In the graph of figure 3, the source node is s and the destination node is d. The parameter X is set equal to $1/2 \sum_{a_i \in A} s(a_i)$.

It is easy to see that the construction of the instance of the ND2PP problem can be accomplished in polynomial time. We need to show that instance of the ND2PP problem will have two node disjoint paths from s to d of length at most X, if and only if elements of the instance of the PartProb can be divided into two groups, such that the sum of the sizes of the elements in one group is equal to the sum of the size of the elements of the other group.

Suppose that the elements of the set A can be divided into A' and $A - A'$ such that $\sum_{a_i \in A'} s(a_i) = \sum_{a_i \in A - A'} s(a_i)$. In this case we can construct two node disjoint paths from s to d such that the length of the longest path is at most $1/2 \sum_{a_i \in A} s(a_i)$. Suppose that the set A' contains m elements of the set A ($A = \{a_1, \ldots, a_n\}$, and they are given by $A' = \{a_{f(1)}, a_{f(2)}, \ldots, a_{f(m)}\}$, where $f : \{1, \ldots, m\} \rightarrow \{1, \ldots, n\}$ is a mapping function. We also assume $f(1) \leq f(2) \leq \ldots \leq f(m)$.

The paths P_1 and P_2 may be constructed in the following way:

$P_1 : s = u_0 \rightarrow u_1 \rightarrow \ldots \rightarrow u_{f(1)-1} \rightarrow v_{2f(1)-1} \rightarrow v_{2f(1)} \rightarrow u_{f(1)+1} \rightarrow \ldots \rightarrow u_{f(2)-1} \rightarrow v_{2f(2)-1} \rightarrow v_{2f(2)} \rightarrow u_{f(2)+1} \rightarrow \ldots \rightarrow u_{f(m)-1} \rightarrow v_{2f(m)-1} \rightarrow v_{2f(m)} \rightarrow u_{f(m)+1} \rightarrow u_{f(m)+2} \rightarrow \ldots \rightarrow u_n \rightarrow u_{n+1} = d$

$P_2 : s = v_0 \rightarrow v_1 \rightarrow \ldots \rightarrow v_{2f(1)-2} \rightarrow u_{f(1)} \rightarrow v_{2f(1)+1} \rightarrow v_{2f(1)+2} \rightarrow \ldots \rightarrow v_{2f(2)-2} \rightarrow u_{f(2)} \rightarrow v_{2f(2)+1} \rightarrow \ldots \rightarrow v_{2f(3)-2} \rightarrow u_{f(3)} \rightarrow v_{2f(3)+1} \rightarrow \ldots \rightarrow v_{2f(m)-2} \rightarrow u_{f(m)} \rightarrow v_{2f(m)+1} \rightarrow \ldots \rightarrow v_{2n} \rightarrow v_{2n+1} = d$

It can be easily verified that the length of the paths P_1 and P_2 will both be $X = 1/2 \sum_{a_i \in A} s(a_i)$.

Now suppose that the instance of ND2PP has two different node disjoint paths P_1 and P_2 from s to d such that the length of the longer path P_2 is at most $X = 1/2 \sum_{a_i \in A} s(a_i)$. Consider the directed edges $v_{2i-1} \rightarrow v_{2i}, 1 \leq i \leq n$ of the graph, each with weight a_i associated with it. Since the paths P_1 and P_2 are node disjoint, such an edge, $v_{2i-1} \rightarrow v_{2i}, 1 \leq i \leq n$, cannot be a part of both P_1 and P_2. Suppose, if possible such an edge is not part of either P_1 or P_2. Because of the structure of the graph, if both P_1 and P_2 (establishing a path between the nodes s and t) are not using the edge $v_{2i-1} \rightarrow v_{2i}, 1 \leq i \leq n$, then both of them must be passing through the node $u_i, 1 \leq i \leq n$, contradicting the assumption that the paths P_1 and P_2 are node disjoint. Therefore, each edge $v_{2i-1} \rightarrow v_{2i}, 1 \leq i \leq n$ (with a weight a_i associated with it) must be part of either P_1 or P_2. Since the weights associated with every other edge is zero, the length of the path is determined only by the weights of these edges. Since the sum of the weights of these edges is $\sum_{a_i \in A} s(a_i)$, the sum of the path lengths of P_1 and P_2 must be equal to $\sum_{a_i \in A} s(a_i)$. Since the length of the longer path P_2 is equal to $X = 1/2 \sum_{a_i \in A} s(a_i)$, the length of the shorter path P_1 is also equal to $X = 1/2 \sum_{a_i \in A} s(a_i)$. Therefore, the sum of the weights associated with the edges of the path P_1 is equal to the sum of the weights associated with the edges of the path P_2. This implies that the subset of the weights $v_{2i-1} \rightarrow v_{2i}, 1 \leq i \leq n$, associated with the edges of the path P_1 add up to $X = 1/2 \sum_{a_i \in A} s(a_i)$ and the remaining weights of the set $v_{2i-1} \rightarrow v_{2i}, 1 \leq i \leq n$ are associated with the edges of the path P_2. This proves that if there exists two node disjoint paths P_1 and P_2 from s to t, then the elements of the set $A = \{a_1, a_2, \ldots, a_n\}$ can be partitioned into two subsets, such that the sum of the sizes of the elements in one set is equal to the sum of the sizes of the elements of the other set. This proves the theorem.

2.2 Edge Disjoint Path Problem

Edge Disjoint k-Path Problem - *Optimization Version*

Instance: A graph $G = (V, E)$, with weight $w_{i,j}$ associated with each edge $e_{i,j}$, $(w_{i,j} \geq 0)$, a source node s, a destination node d and a positive integer k.

Problem: Find a set of k edge-disjoint paths from s to d such that the length of the *longest* path in this set is the *shortest* among all such sets of k edge-disjoint paths from s to d.

Edge Disjoint k-Path Problem (EDkPP) - *Decision Version*

Instance: A graph $G = (V, E)$, with weight $w_{i,j}$ associated with each edge $e_{i,j}$, $(w_{i,j} \geq 0)$, a source node s, a destination node d and a positive integer k and a non-negative number X.

Question: Is there a set of k edge-disjoint paths from s to d such that the length of the *longest* path in this set is less than or equal to X ?

Theorem 2 *The edge disjoint k-path problem is* NP-complete.

Proof: We show EDkPP is NP-complete by considering a restricted version of the problem where $k = 2$. We call the restricted version the *Edge Disjoint* 2*-Path Problem (ED2PP)*.

Edge Disjoint 2-Path Problem (ED2PP)

Instance: A graph $G = (V, E)$, with weight $w_{i,j}$ associated with each edge $e_{i,j}$, $(w_{i,j} \geq 0)$, a source node s, a destination d and a non-negative number X.

Question: Is there a set of two edge-disjoint paths from s to d such that the length of the *longest* path in this set is less than or equal to X ?

We show ED2PP is NP-complete by giving a transformation from the ND2PP problem discussed in the previous section.

Proof: It is easy to see that ED2PP $\in$ NP, since a non-deterministic algorithm need only guess two edge disjoint paths P_1 and P_2 and check in polynomial time that the length of the longer path is less than or equal to X.

We will transform ND2PP to ED2PP. We construct an instance of the ED2PP from an instance of the ND2PP by replacing each node v_i of the instance of ND2PP with two nodes $v_{i,1}$ and $v_{i,2}$. A directed edge is introduced from $v_{i,1}$ to $v_{i,2}$ and a weight of 0 is assigned on this edge. It is easy to check that the instance of ED2PP will have two edge-disjoint paths from s to d such that the length of the *longer* path among the two paths, is less than or equal to X if and only if the instance of ND2PP will have two node-disjoint paths from s to d such that the length of the *longer* path is less than or equal to X.

3 Mathematical Programming Formulation

A mathematical programming [14] solution to the optimization version of the EDPP is given next. This formulation seeks to find a set of k edge disjoint paths from the source to the destination such that the length of the *longest* path in this set is *shortest* among all such sets of paths. If only one *backup* is needed, as is the case in the WDM protection scheme, the parameter k is set equal to 2. If k edge disjoint paths are maintained ($k \geq 2$), it can tolerate up to $k-1$ failures.

The node disjoint version of the problem can be transformed to the edge disjoint version in $O(n)$ time, where n, is the number of nodes in the network through a technique known as *node splitting* [30]. In this technique, every node v_i in the graph is replaced with two nodes v_{i1} and v_{i2} and directed edges, $v_{i1} \rightarrow v_{i2}$ and $v_{i2} \rightarrow v_{i1}$.

The variable r is used to identify a path, $1 \leq r \leq k$. A binary indicator variable $x^r_{i,j}$ is associated with each link (i,j) of the directed graph $G = (V, E)$. If the variable $x^r_{i,j} = 1$, it indicates that the link (i,j) is a part of the r-th path from the source to the destination and if $x^r_{i,j} = 0$ then it is not a part of such a path.

Minimize C

Subject to the following constraints:

$$(i) \sum_{\{i|i\in V\}} x^r_{i,j} - \sum_{\{i|i\in V\}} x^r_{j,i} = \begin{cases} -1 & \text{if } j \text{ is the source node} \\ 1 & \text{if } j \text{ is the destination node} \\ 0 & \text{if } j \text{ is any other node} \end{cases}$$

$$(ii) \sum_{(i,j)\in E} w_{i,j}x^r_{i,j} \ \leq \ C, \ \forall \ r, 1 \leq r \leq k$$

$$(ii) \sum_{r=1}^{k} x^r_{i,j} \ \leq \ 1, \ \forall \ (i,j) \in E$$

$$(iii) \ x^r_{i,j} \ = \ 0/1, \ \forall \ r, \ 1 \leq r \leq k; \ \forall (i,j) \in E$$

An exact solution to the EDPP can be obtained by solving the integer linear programming problem given above. However, the computational time may be high if the problem size is large. Next, we give a polynomial time algorithm that finds an approximate solution with a guaranteed performance bound for the NP-complete EDPP problem.

4 Approximate solution to the Path Problem

Consider another edge disjoint path problem with a different objective:

Edge Disjoint k-Path Problem - Minimum Total Edge Cost (EDkPP-MTEC)

Instance: A graph $G = (V, E)$, with weight $w_{i,j}$ associated with each edge $e_{i,j}$, ($w_{i,j} \geq 0$), a source node s, a destination node d and a positive integer k.

Problem: Find a set of k edge-disjoint paths from s to d such that the *sum* of the length of the paths in this set, is the *minimum* among all such sets of k edge-disjoint paths from s to d.

Clearly, EDkPP is different from EDkPP-MTEC in its objective. The computational complexities of these problems are also different. In this paper, we have shown that the

EDkPP is NP-complete, whereas it is well known that the EDkPP-MTEC can be solved in polynomial time [30, 31]. If n is the number of nodes in the graph and m is the number of edges, then the set of k edge-disjoint paths from a given source s to a given destination d, with minimum total edge cost can be found in $O(km\log_{1+m/n} n)$ time [15, 31].

We use the algorithm to compute the optimal solution to the EDkPP-MTEC problem [15] as an approximate solution to the EDkPP problem. Suppose that the k disjoint paths generated by the optimal solution of the EDkPP-MTEC problem are $P_1, P_2, \ldots P_k$ and the optimal solution of the EDkPP are the k disjoint paths $P'_1, P'_2, \ldots P'_k$. We denote the length of the paths P_i and P'_i by PL_i and PL'_i respectively. Without loss of generality, we assume that $PL_i \leq PL_{i+1}$ and $PL'_i \leq PL'_{i+1}$, for $1 \leq i \leq k-1$.

We measure the quality of the approximate solution of the EDkPP problem by the ratio PL_k/PL'_k.

Since the path set $P_1, P_2, \ldots P_k$ is the optimal solution to the EDkPP-MTEC problem,

$$PL_1 + PL_2 + \ldots + PL_k \leq PL'_1 + PL'_2 + \ldots + PL'_k$$

Since $PL'_i \leq PL'_{i+1}$ for $1 \leq i \leq k-1$, the above equation can be rewritten as

$$PL_1 + PL_2 + \ldots + PL_k \leq kPL'_k$$

or $PL_k \leq kPL'_k$ as $PL_i \geq 0$ for $1 \leq i \leq k$. This reduces to

$$PL_k/PL'_k \leq k$$

Thus the ratio of the *approximate solution* to the EDkPP, PL_k, to the *optimal solution*, PL'_k, is bounded by k.

In the protection scheme in the lightwave network only two disjoint paths (one primary and one backup) are sought, i.e., $k = 2$. Therefore, in this case the approximate solution is bounded by 2 times the optimal solution. Closer examination reveals that the ratio

PL_2/PL'_2 must be *strictly less* than 2. Suppose, if possible $PL_2 = 2PL'_2$. As $PL_1+PL_2 \leq PL'_1 + PL'_2$, this implies $PL_1 + 2PL'_2 \leq PL'_1 + PL'_2$ or $PL'_2 \leq PL'_1 - PL_1$. This cannot be true unless $PL_1 = 0$, because from our initial assumption we have, $PL'_1 \leq PL'_2$. Therefore the PL_2 must be less than $2PL'_2$.

Next we show that this bound is *tight*. Consider the example shown in figure 2. The MinSum path pair in this example are $A \to D \to C \to B$ and $A \to B$ with a total path length $2x - \epsilon$ and the MinMax path pair in this example is $A \to C \to B$ and $A \to D \to B$ with a total path length $2x$. The length of the backup path in the first case is $2x - 4\epsilon$ whereas in the second case it is only x. Thus the ratio between the backup path length of the MinSum pair to the backup path length of the MinMax pair may be very close to 2.

4.1 Algorithm for Edge Disjoint k-Path (Minimum Total Edge Cost) Problem

In this subsection, we briefly sketch the algorithm we use to compute MinSum Path Pair (MSPP).

The MSPP is a special case of the *minimum cost flow problem* [15]. The minimum cost flow formulation of the MSPP given below is from [1]:

A binary indicator variable $x_{i,j}$ is associated with each link (i,j) of the directed graph $G = (V,E), |V| = n, |E| = m$. If the variable $x_{i,j} = 1$, indicates that the link (i,j) is a part of the path from the source to the destination and $x_{i,j} = 0$ otherwise. The parameter $b(i)$ associated with each node $i \in V$, indicates the supply or demand at the node i and $c_{i,j}$ is the cost of the edge (i,j).

Minimize $\sum_{(i,j)\in E} c_{i,j}x_{i,j}$

Subject to the following constraints:

$$(i) \sum_{\{j|(i,j)\in E\}} x_{i,j} - \sum_{\{j|(j,i)\in E\}} x_{j,i} = b(i) \quad \forall i \in V$$

$$(ii) x_{i,j} = 0/1 \forall (i,j) \in E$$

Lawler in [15] has given a minimum cost augmenting path algorithm for the minimum cost network flow. Suurballe in [30] present an algorithm for the same problem for *node-disjoint* paths. An efficient implementation of the edge-disjoint path problem with minimum total edge cost is given in [31]. As we use it to compute the MSPP in the graph, we briefly describe the algorithm from [31] .

The algorithm performs two preliminary steps:

Step 1: It finds a shortest path tree T rooted at s. For any node v of the graph, this tree contains a shortest path from s to v. The edges of this tree are referred to as the *tree edges* and the other edges as the *non-tree edges*. The shortest path distance from the node s to any node v in the graph is denoted by $d(s,v)$.

Step 2: The length of each edge (v,w) denoted by $c(v,w)$ is transformed by defining $c'(v,w) = c(v,w) + d(s,v) - d(s,w)$.

For any node v, G_v denotes the graph formed from G by reversing all the edges along the path T from s to v.

The following theorem stated in [31] follows from the minimum cost augmenting path algorithm for the minimum cost network flow problem given in [15].

Theorem 3 *For any vertex v, the total length of a shortest pair of edge disjoint paths from s to v, is $d_v(s,v)$, where d_v denotes the distance function in G_v. Such a pair of*

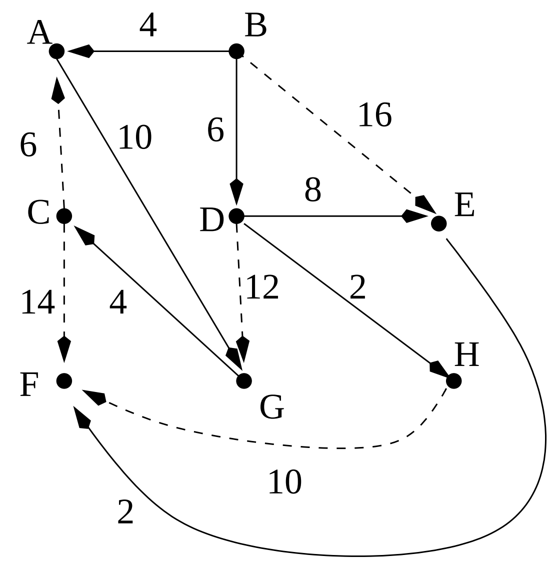

Figure 4: Shortest Path Tree

paths can be obtained from the path from s to v in T and a shortest path from s to v in G_v by taking the union of the edges in the two paths, discarding every edge in one path whose reversal appears in the other, and grouping the remaining edges into two paths.

The execution of the algorithm is demonstrated with the help of an example shown in figure 4. In the example, B and F are the source and the destination nodes respectively. First a shortest path tree rooted at the source node B is constructed. The tree edges are shown as solid lines in the figure 4. The figure 5 shows the edges weights after the transformation mentioned in step 2 earlier. In figure 6, the direction of the shortest path from B to F in figure 4, $B \rightarrow D \rightarrow E \rightarrow F$, is reversed. The shortest path from B to F in the graph of figure 6 is computed and that path is $B \rightarrow E \rightarrow D \rightarrow H \rightarrow F$. Using theorem 1 we obtain the pair of disjoint paths whose sum of the path lengths is the smallest as $B \rightarrow E \rightarrow F$ and $B \rightarrow D \rightarrow H \rightarrow F$.

The MSPP problem can also be solved using a distributed algorithm given in [20].

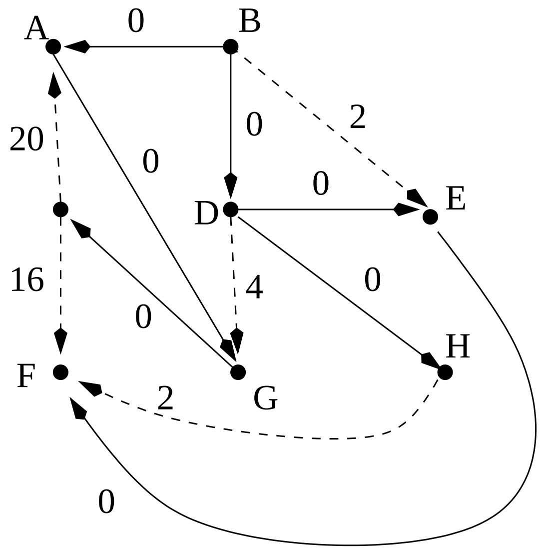

Figure 5: Graph after transformation of edge weight

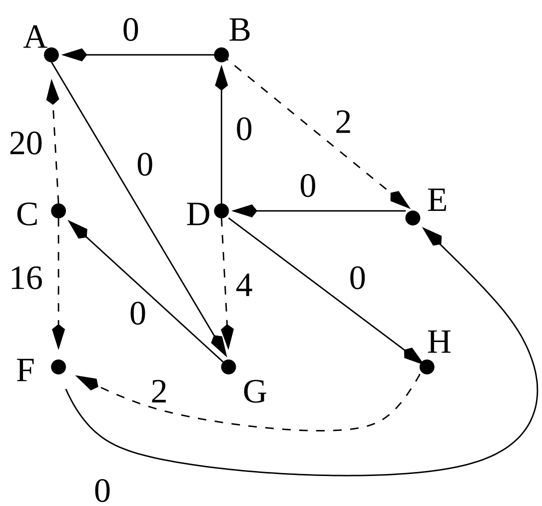

Figure 6: Graph after reversal of shortest path

Graph No.	No. of Nodes	Avg. Degree (approx)	MinMaxPP		MinSumPP		SPP	
			PP	SP	PP	SP	PP	SP
1	20	15	6	6	5	6	5	7
2	25	8	9	9	8	9	8	9
3	40	6	13	13	12	13	12	13
4	40	20	6	6	4	6	4	6
5	55	4	21	26	21	26	18	30
6	85	4	19	19	14	20	14	20
7	85	30	6	6	5	6	5	6
8	100	6	24	24	23	24	23	24
9	100	18	9	9	6	11	6	11
10	100	50	3	3	2	3	2	3

Table 1: Comparison of MinMax, MinSum and SPP; PP: Primary Path, SP: Secondary Path

5 Comparison of Exact and Approximate Solution

In this section we present the results of our experimental evaluation of three problems, (i) Short Path Pair, (ii) MinMax Path Pair and (iii) MinSum Path Pair. We first created a random graph generator. Taking the number of nodes and the average node degree as input parameters, the generator creates a random graph. We find the Shortest Path Pair (SPP) by executing Dijkstra's shortest path algorithm twice, once on the input graph and once on the input graph after removal of the edges belonging to the shortest path. As described in the previous section, the MinSum Path Pair (MinMaxPP) is also computed by execution of the Dijkstra's algorithm twice. The MinSum Path Pair (MinMaxPP) is computed by using the mathematical programming formulation given in section 3. The integer linear program is executed on a SUN Ultra workstation using CPLEX 6.5 mathematical programming package. The computation time for the problems using the CPLEX was less than a minute. The lengths of the primary and the secondary paths, obtained by MinSumPP, MinMaxPP is presented in the table 1.

In this paper we have proven that the backup path length in the MinSum pair will always be less than two times the backup path length in the MinMax pair. Our experimental results presented in the table 1 shows that the performance of MinSum pair is even superior. In our experiments the maximum value of the ratio of the backup path length in the MinSum pair to the backup path length in the MinMax pair was 1.2.

It may observed that the results of the Shortest Path pair is very close to the results of the MinSum pair. Since the computational complexity of the SP pair is identical to that of the MinSum pair, this may suggest the use SP pair instead of the MinSum pair. However, there is a significant difference between the two. The MinMax pair can *guarantee* that the length of its backup path will always be less than two times the backup path length of the MinMax pair. However, the SP pair cannot provide any such guarantee. We illustrate this with an example. In the graph of figure 7, S and D are the source and the destination nodes respectively. The weights of the links is shown in the figure. The SP pair from S to D in this graph is $S \rightarrow A \rightarrow B \rightarrow C \rightarrow E \rightarrow F \rightarrow G \rightarrow D$ and $S \rightarrow L \rightarrow M \rightarrow D$ with path lengths 7ϵ and $y + 2\epsilon$ respectively, and the MinMax pair is $S \rightarrow J \rightarrow E \rightarrow F \rightarrow G \rightarrow D$ and $S \rightarrow H \rightarrow B \rightarrow C \rightarrow I \rightarrow K \rightarrow D$ with path lengths $x + 4\epsilon$ and $x + 5\epsilon$ and respectively (x, y are any arbitrary numbers and ϵ is an arbitrarily small number). If y is arbitrarily large in comparison with x, the ratio of the backup path length in the SP pair $(y + 2\epsilon)$ to the backup path length in the MinMax PP $(x + 5\epsilon)$ may be arbitrarily large.

Our experiments show that the results produced by the MinSum pair is very close to the MinMax pair. It is also close to the SP pair. Since the computational complexity of the MinSumPP is identical to that of the SPP and the MinSumPP *can provide a performance guarantee* while the SPP *cannot*, there is no compelling reason for using the SPP instead of the MinSumPP as the primary and backup paths in optical networks.

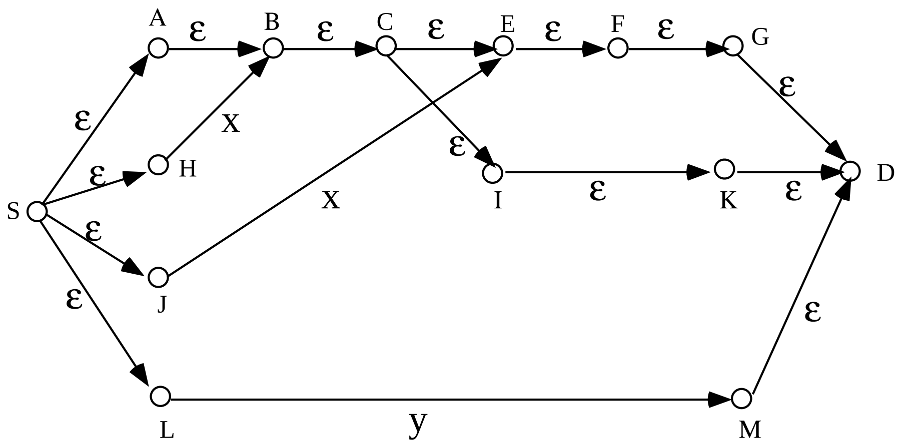

Figure 7: Comparison of Shortest Path pair and MinMax pair

6 Variations of Disjoint Path Problem

In the previous sections, we have discussed the problem of finding a set of k edge disjoint paths between a source-destination pair so the length of the longest path is shortest among all such sets. In WDM networks we are particularly interested in the case $k = 2$, where the two disjoint paths correspond to the primary and backup paths respectively. The techniques presented earlier attempt to minimize the length of the backup path length (the length of the primary path is always shorter than the length of the backup path). In the example shown in figure 2, we have the option of choosing a primary path of length 3ϵ and a secondary path of length $2x - 4\epsilon$ or a primary and a secondary path, both of whose lengths are x. In this paper we have argued that the second choice is a *better* choice, because the length of the backup path in the first choice may be very long resulting in an unacceptably high delay. This may be very critical in an QoS constrained environment. However, the length of the backup path is important only if we assume that there are occasions when the primary path is not available and the secondary path has to be used for data transmission. The arguments to support the claim that the first choice is a better choice is based on the assumption that the reliability of the network is very high and the backup path, if it is at all used, will be used very rarely. In such a situation, attention should be paid to *minimize* the *length of the primary path* and not the *length of the secondary path*, as the primary path is the one responsible for data transfer for *most of the time*.

In this section we provide a formulation that minimizes *combined primary and secondary* path length. We refer to this problem as the *Combined Primary Secondary Path Pair, CPSPP* computation problem. Suppose p is the probability of failure of the primary path. We are assuming a single fault scenario, which implies that if the primary path is not operational due to a fault, the secondary (or backup) path is operational (faultfree). In this case, we minimize a combined path length which is equal to

$$(1-p) * (primary\ path\ length) + p * (secondary\ path\ length)$$

The variable r is used to identify a path, $1 \leq r \leq 2$ ($r = 1$ corresponds to the primary path and $r = 2$ corresponds to the backup path). A binary indicator variable $x^r_{i,j}$ is associated with each link (i,j) of the directed graph $G = (V, E)$. If the variable $x^r_{i,j} = 1$, it indicates that the link (i,j) is a part of the r-th path from the source to the destination and if $x^r_{i,j} = 0$ then it the link (i,j) is not a part of such a path.

Minimize $\sum_{(i,j)\in E}[(1-p)*w_{i,j}x^1_{i,j} + p*w_{i,j}x^2_{i,j}]$

Subject to the following constraints:

$$(i) \sum_{\{i|i\in V\}} x^r_{i,j} - \sum_{\{i|i\in V\}} x^r_{j,i} = \begin{cases} -1 & \text{if } j \text{ is the source node} \\ 1 & \text{if } j \text{ is the destination node} \\ 0 & \text{if } j \text{ is any other node} \end{cases}$$

$$(ii) \sum_{(i,j)\in E} w_{i,j}x^1_{i,j} \leq \sum_{(i,j)\in E} w_{i,j}x^2_{i,j}$$

$$(iii) \sum_{r=1}^{2} x^r_{i,j} \ \leq \ 1, \ \forall \ (i,j) \in E$$

$$(iv) \ x^r_{i,j} \ = \ 0/1, \ \forall \ r, \ 1 \leq r \leq 2; \ \forall (i,j) \in E$$

The CPSPP is a generalization of SPP, MSPP and MMPP problem. With appropriate choice of the parameter p, and trivial modification of the formulation, all three of them can be computed using the above set of constraints and objective function.

7 Conclusion

In this paper we have proposed the use of the MinMax Path Pair (MMPP) as the primary and the backup path in the protection scheme of WDM networks. We have provided a rationale for using the MMPP instead of the currently practiced the Shortest Path Pair (SPP) as the primary and the backup paths. We have proved that selection of the MMPP is an NP-complete problem both in its edge-disjoint and node-disjoint versions. We have proposed the use of the MinSum Path Pair (MSPP) as an approximation of the MMPP. Through analysis and simulation, we have shown that the MSPP very closely emulates the MMPP. In many of the experimental cases, the performance is identical. Since the computational complexity of the MSPP is very low (it is identical to the complexity of the SPP), and the MSPP *can provide a performance guarantee* while the SPP *cannot*, there is no compelling reason for using the SPP instead of the MSPP as the primary and backup paths in optical networks. We also provide a formulation of the Combined Primary Secondary Path Pair (CPSPP), which is a generalization of the notions of SPP, MMPP and MSPP.

References

[1] R. K. Ahuja, T. L. Magnanati and J. B. Orlin, Network Flows: Theory, Algorithms and Applications, Prentice Hall, 1993.

[2] S. Bandyopadhyay, A. Sengupta and A. Jaekel, "Fault-tolerant routing scheme for All Optical Networks", *Proc. SPIE All Optical Networking Conf: Architecture, Control and Management Issues*,pp. 420-431, Boston, MA, Nov. 1998.

[3] R. Bhandari, "Optimal diverse routing in telecommunication fiber networks", *Proc. IEEE INFOCOM'94*, pp. 1498-1508, 1994.

[4] B. Doshi et. al., "Optical network design and restoration", Bell Laboratories Technical Journal, vol. 4,no. 1, pp. 58-84, Jan-Mar 1999.

[5] D. A. Dunn, W. .D. Grover, M. H. MacGregor, "Comparison of k-shortest paths and maximum flow routing for network facility restoration", *IEEE Selected Areas of Communications* , vol. 12, no. 1, 1994.

[6] P.K. Dowd, "Wavelength division multiple access channel hypercube processor interconnection", *IEEE Transactions on Computers*, 1992.

[7] S. G. Finn *et. al.*, "A novel approach to automatic protection switching using trees", *Proc. IEEE ICC'97*, pp. 272-276, 1997.

[8] A. Fumagalli *et.al*, "Survivable networks based on optimal routing and WDM self healing rings", Proc. of IEEE INFOCOM'99, pp. 726-733, March, 1999.

[9] M.R. Garey and D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman Publishers, San Francisco, 1978.

[10] O. Gerstel and R. Ramaswami, "Optical layer survivability: A services perspective", *IEEE Communications Magazine*, vol. 38, no. 3, pp. 104-113, March 2000.

[11] N. Ghani, S. Dixit and T. S. Wang,"On IP-over-WDM Integration", *IEEE Communications Magazine*, vol. 38, no. 3, pp. 72-84, March 2000.

[12] N. Ghani, S. Dixit and T. S. Wang,"Channel provisioning for higher layer protocols in WDM networks", *Proc. SPIE All Optical Networking Conf: Architecture, Control and Management Issues*, Boston, MA, Sept. 1999.

[13] M.G. Hluchyj and M.J. Karol, "Shufflenet: An application of generalized perfect shuffles to multihop lightwave networks", *Journal of Lightwave Technology*, vol. 9, no. 10, October, 1999.

[14] T. C. Hu, Integer Programming and Network Flows, Addison Wesley Publishing Company, 1970.

[15] E. L. Lawler, Combinatorial Optimization: Networks and Matroids. Holt, Rinehart and Winston, New York, 1976.

[16] M. A. Marsan, A. Bianco,E. Leonardi and F. Neri, "Topologies for Wavelength-Routing All-Optical Networks", *IEEE/ACM Transactions on Networking*, vol. 1, pp. 534-546, 1993.

[17] M. F. Maxemchuk, "Regular mesh topologies in local and metropolitan area networks", *AT & T Technical Journal*, vol. 64, pp. 1659-1686, September, 1985.

[18] M. Medard *et. al.*, "Redundant trees for automatic protection switching in arbitrary node-redundant or edge-redundant trees", *IEEE/ACM Trans. Networking* (to appear), 1999.

[19] S. D. Nikolopoulos, A. Pitsilides and D. Tipper,"Addressing network survivability issues by finding the k-best paths through a Trellis graph", *Proc. IEEE INFOCOM'97*, pp. 370-377, 1997.

[20] R. G. Ogier, V. Rutenburg and N. Shacham,"Distributed algorithms for computing shortest pair of disjoint paths", *IEEE Trans. Information Theory*, vol. 39, no. 2, pp. 443-455, March 1993.

[21] G. Panchapakesan and A. Sengupta, "On multihop optical network topology using Kautz digraph," *Proc. IEEE INFOCOM'95*, pp. 675-682, April, 1995.

[22] S. Ramamurthy and B. Mukherjee, "Survivable WDM Mesh Netwoks", Proc. of IEEE INFOCOM'99, pp. 744-751, March, 1999.

[23] L. Sahasrabuddhe, S. Ramamurthy and B. Mukherjee,"Fault management in IP-over-WDM networks: WDM protection vs. IP restoration", Ph.D Thesis, University of California, Davis, 1999.

[24] A. Sen and P. Maitra, "Supercube: AN optimally fault-tolerant network architecture for lightwave applications", *Computer Networks and ISDN Systems*, pp., 1994.

[25] A. Sen, T. Shah, S. Bandyopadhyay and B.P. Sinha, "On new architectures for WDM networks", *Proc. of 37th Annual Allerton Conference on Communication, Control and Computing*, pp. 402-413, University of Illinois at Urbana-Champaign, September 1999.

[26] A. Sen, S. Bandyopadhyay and B.P. Sinha, "A new architecture for Lightwave networks", To be presented at the 2000 *Symposium on Performance Evaluation of Computer and Telecommunication Systems*,to be held in Vancouver, Canada in July 2000.

[27] K. Sivarajan and R. Ramaswami, "Lightwave networks based on de Bruijn graphs," *IEEE/ACM Transactions on Networking*, vol. 2, no. 1, pp. 70-79, 1994.

[28] C. C. Skiscim and B. L. Golden, "Solving k-shortest and constrained shortest path problems efficiently", *Annals of Operations Research*, vol. 20, pp. 249-282, 1989.

[29] T. E. Stearn and Krishna Bala, Multiwavelength Optical Networks - A Layered Approach, Addison Wesley, 1999.

[30] J. W. Suurballe, "Disjoint paths in a network", *Networks*, vol. 4, pp. 125-145, 1974.

[31] J. W. Suurballe and R. E. Tarzan, "A quick method for finding the shortest pair of disjoint paths", *Networks*, vol. 14, pp. 325-336, 1984.

[32] K.W. Tang, "CayleyNet: A multihop WDM-based lightwave network," *Proc. IEEE INFOCOM'94*, June, 1994.