Robust Routing in Wide-Area WDM Networks

Weifa Liang
Department of Computer Science
Australian National University, Canberra, ACT 0200, Australia

Abstract

This paper considers the problem of establishing robust routes for user connection requests in an WDM network dynamically. The problem is to find two edge-disjoint routes with satisfying certain given properties. One route will serve as the primary path, and another will serve as the backup path which will replace the primary path if there is any link failure in the primary path. Two versions of the problem are studied: one is to find two edge-disjoint paths such that the total cost of the two paths is minimized, in terms of the network resources consumption; the other is to find two edge-disjoint paths to minimize both the network load (link congestion) and the total cost of the two paths. The exact and approximate algorithms for the problem are proposed, and the solutions delivered consist of selecting routes, assigning wavelengths to the links, and setting switches of wavelength conversion at intermediate nodes on the routes. The performance ratio between the approximate solution and the exact solution is also analyzed. The key technique used in the design of the approximate algorithms, is to transform the corresponding version into a well solved optimization problem on an auxiliary graph. To the best of our knowledge, this is the first time that in the design of routing protocols for WDM networks, the network load and the route finding and wavelength assignment are taken into account simultaneously. As results, it not only finds cheap routes but also reduces the number of network re-configurations, thereby improving the performance of the network through utilizing its resources effectively.

1 Introduction

All-optical networks employing wavelength division multiplexing and wavelength routing are a promising candidate for future WANs. These networks offer the advantages of wavelength reuse and scalability. Wavelength division multiplexing (WDM) divides the bandwidth of an optical fiber into multiple wavelength channels, so that multiple users can transmit their data at distinct wavelengths through the same fiber concurrently [6, 9, 14]. The large bandwidth of optic fibers have made them attractive for high-speed networks. The major applications of the network are video conferencing, scientific visualization, real-time medical imaging, supercomputing, and distributed computing [2, 20, 22]. But the existence of many independent data channels over a single fiber could lead to problems in case of failures, as the amount of bandwidth lost by a component or a link failure is now much larger than what would have been lost in a traditionally electronic network. Therefore, the study of robust routing is a fundamental topic in the WDM optical networks because the performance of a semilightpath/lightpath in WDM networks is determined by the quality of the optical components along the path, the failures such as fiber cuts and wavelength failures will affect the performance of the entire network in the end. This is why an extra effort is being spent in analyzing optical WDM networks and finding ways to protect them against failures [8, 23, 4, 11]. Generally, there are two approaches to deal with the failure issue. One is based on hardware protection, which aims to ensure a biconnectional communications channel between two nodes will remain operational in the event of a component or an optical link failure. This protection is performed at the physical level and through the hardware duplication. The cost of hardware protection and the guarantee of connectivity between network nodes may be prohibitively high for its implementation, and thus in the real world, it is only found in the mission-critical business routes. The other is based on software protection which aims to find backup paths for each primary path such as the backup paths and the primary path are either edge-disjoint (protecting a single link failure) or node-disjoint (protecting single node and single link failures). In this paper we shall focus on the software protection by considering the single-link failure. In particular we shall concentrate on establishing the connection with specified failure restoration guarantees in an WDM network with dynamic traffic demands.

In a network with dynamic traffic demand, user connection requests arrive to and depart from the network in a random manner. In response to new requests, semilightpaths are established dynamically. The robust routing for dynamic traffic has been addressed for non-WDM networks such as ATM networks [10, 11, 15]. When a link or a node fails, how to re-establish a new route to backup the failed primary route is very important. The methods of recovering from such failure can be classified into *activate* and *passive* approaches. The passive approach is the simplest way of recovering from failures in which a new connection that does not used the failed component is selected and established if available. The advantage of this method is low overhead in the absence of failures. However, this does not guarantee the successful recovery, as the attempt to establish a new connection may fail due to resource shortage at the time of failure recovery. To overcome the drawbacks of the passive approach, the activate approach is to establish a backup path dynamically during establishing the primary path for a given connection request. Since the backup connection is established and reserved before any failure actually occurs. It can be used immediately upon occurrence of a failure in the primary path without invoking the time-consuming connection re-establishment process. Hence, the failure recovery delay of the activate method is much smaller, leading to fast recovery. Some variants of the activate method have also been suggested which can been seen in [11].

Some recent effort for the robust routing in optical networks has also been taken [11, 17, 16, 15]. In [17] the mechanisms to detect and isolate faults such as fiber cuts and router failures have been established. The problem of fault-tolerant design for the static traffic demand has also been addressed in [17, 3]. These algorithms can afford to be computationally expensive as they run off-line. On the other hand, the dynamic routing schemes must be simpler and faster algorithms because short-lived connections are setup and torn down frequently. Several dynamic routing algorithms have also been proposed [16, 15, 11]. In [16] heuristics are given to establish routes for several concurrent connection requests with the goal to reduce the blocking probabilities of the entire network. In [11] an algorithm for finding the primary and backup paths for the Multi-Protocol-Level-Switching (MPLS) is proposed which is claimed to be applicable to find routing wavelength paths for optical networks. The all above algorithms, however, are assumed that the optical network concerned is an undirected graph, and do not take the costs of switching setting and of traversing links using wavelengths into consideration, which we will focus on in this paper. Therefore, their algorithms are not suitable for our case which is more complicated than theirs.

To design an efficient routing protocol, at least two factors must be considered. One is to find economical routes for user connection requests in terms of the cost; another is to alleviate the number of network reconfigurations while the network reconfiguration aims to minimize the maximum link load [18, 1] by resetting the routes to alleviate the traffic congestion on some heavily loaded links. In all previously known routing algorithms, the reconfiguration and the route finding and wavelength assignment (RWA) are treated as separate tasks. Researchers only either consider finding efficient algorithms to minimize the number of wavelengths used and/or routing delay time or focus on finding an optimal time point to perform the reconfiguration. Note that during the reconfiguration period, the network is in the frozen mode and does not respond to any user connection request and perform the routing.

Our major contributions in this paper are as follows. The robust routing problem on an WDM network is defined. Exact and approximate algorithms for the problem are presented. The performance ratios between the approximate and the exact solutions are also analyzed. The key technique used is to reduce the problem to a well known optimization problem on a corresponding auxiliary graph. To the best of our knowledge, this is the first time that in designing routing protocols for WDM networks, the network load and the route finding and wavelength assignment are considered simultaneously. As results, the chosen routes not only are cost-effective but also can alleviate the number of network re-configurations, thereby improving the performance of the network through utilizing its resources effectively.

2 Preliminaries

The optical network is modeled by a *directed graph* $G = (V, E, \Lambda)$, where V is a set of nodes and E is a set of directed links (edges) in the network, $n = |V|$ and $m = |E|$. Let $\Lambda = \{\lambda_1, \lambda_2, \dots, \lambda_W\}$ be the wavelength set in G. Associated with each node $v \in V$, there is a switch converter which can convert an incoming wavelength to an outgoing wavelength if necessary. The switching operation at a node uses a wavelength conversion table, which is given in advance. Associated with each link $e \in E$, there is a set $\Lambda(e)$ $(\subseteq \Lambda)$ of wavelengths in which one of the wavelengths will be used when traversing e.

Following Chlamtac et al [5], the *cost* of using the resources in $G(V, E, \Lambda)$ is defined as follows. For each link $e \in E$ and a wavelength $\lambda_i \in \Lambda(e)$ a nonnegative weight $w(e, \lambda_i)$ is associated, representing the "cost" of using λ_i on e. The "cost" of wavelength conversion is modeled via cost factors of the form $c_v(\lambda_p, \lambda_q)$, which is the cost of wavelength conversion at node v from λ_p to λ_q. If $\lambda_p = \lambda_q$, then $c_v(\lambda_p, \lambda_p) = 0$. The defined wavelength conversion cost accommodates the general case where the conversion cost depends on nodes and the wavelengths involved.

A *semilightpath* $\mathcal{P}$ in G is a sequence $e_1, e_2, \dots, e_l$ of links such that the tail of e_{i+1} coincides with the head of e_i and a specific wavelength $\lambda_{j_i} \in \Lambda(e_i)$ is associated with e_i, $i = 1, \dots, l$. Denote by $head(e)$ and $tail(e)$ the head and the tail of a directed link e. The *cost* $C(\mathcal{P})$ of $\mathcal{P}$ is thus as follows.

$$C(\mathcal{P}) = \sum_{i=1}^{l} w(e_i, \lambda_{j_i}) + \sum_{i=1}^{l-1} c_{head(e_i)}(\lambda_{j_i}, \lambda_{j_{i+1}}). \quad (1)$$

The optimal semilightpath from a source s to a destination t is such a semilightpath that its cost defined in Eq. (1) is minimized. Finding an optimal semilightpath is different from finding a single source shortest path because, not only do we need to find such an optimal semilightpath, but we also need to assign every link e on the path a specific wavelength $\lambda(e) \in \Lambda(e)$ and set the switch of wavelength conversion at every intermediate node if necessary. Given two semilightpaths from s to t in G, the two paths are *edge-disjoint* if they do not share any physical optic links.

In addition to the number of wavelengths and the time delay on a route are the important network resources, the *network load* is another important network resource, which is defined as follows. Let $U(e)$ be the number of wavelengths on link $e \in E$ which are being used by other routes and $N(e)$ the total number of wavelengths on link e, i.e., $N(e) = |\Lambda(e)|$. Let $\Lambda_{avail}(e)$ be the set of wavelengths having not being used. The *load of link* e is defined as

$$\rho(e) \quad = \quad \frac{U(e)}{N(e)} = \frac{|\Lambda(e)| - |\Lambda_{avail}(e)|}{|\Lambda(e)|}. \qquad (2)$$

The network load ρ is then defined as $\rho = \max\{\rho(e) : e \in E\}$. In the design of routing protocols, to reduce the traffic congestion of routes on some heavily loaded links, an important optimization objective is to minimize the network load ρ.

In this paper we assume that the network accepts user connection requests periodically. At a given time interval, suppose a set of requests is given. The algorithm processes these requests one by one. Once a request is processed and there is a solution for it, the algorithm establishes the routes for it immediately. Otherwise, the request is dropped and a subsequent request is picked up and processed. Formally speaking, given a connection request from s to t, the *robust routing problem* is to establish two edge-disjoint semilightpaths from s to t dynamically such that they satisfy the given property. Two versions of the problem are studied. The first version called the *optimal edge-disjoint semilightpath problem*, is to find two edge-disjoint semilightpaths such that the cost sum of the two paths is minimized. The second version is to find two edge-disjoint semilightpaths such that both the network load and the cost sum of the two paths are minimized simultaneously.

3 The Optimal Edge-Disjoint Semilightpath Problem

In this section we deal with the problem by giving an exact solution for it first, followed by presenting an approximate solution.

3.1 Integer programming solution

We formulate the problem as an integer program. The objective is to determine the primary and backup paths for the connection request considered so as to optimize the use of network resources in terms of the cost. Let the vector $\mathbf{X}$ ($\mathbf{Y}$) of mk coordinates represent the flow on the primary path (the backup path), where the coordinate $x_{ij}^{(l)}$ ($y_{ij}^{(l)}$) is set to 1 if link $\langle v_i, v_j\rangle \in E$ is used in the primary path (backup path) and wavelength λ_l is assigned to it, $1 \le i, j \le n$, $1 \le l \le W$.

The optimal edge-disjoint semilightpath problem is subject to minimize

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{l=1}^{W} x_{ij}^{(l)} w(\langle v_i, v_j\rangle, \lambda_l) + \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} z_{ijk}$$
$$+\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{l=1}^{W} y_{ij}^{(l)} w(\langle v_i, v_j\rangle, \lambda_l) + \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} t_{ijk}. \qquad (3)$$

In Ex. (3), $\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{l=1}^{W} x_{ij}^{(l)} w(v_i, v_j, \lambda_l)$ ($\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{l=1}^{W} y_{ij}^{(l)} w(v_i, v_j, \lambda_l)$) is the cost sum of traversing links on the primary path (backup path); $\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} z_{ijk}$ ($\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} t_{ijk}$) is the cost sum of wavelength switching at the intermediate nodes on the primary path (backup path). The above 0/1 integer programming must satisfy the following constraints.

$$\sum_{l=1}^{W} x_{ij}^{(l)} \le 1, \forall\ \langle v_i, v_j\rangle \in E \qquad (4)$$

$$\sum_{j=1}^{n}\sum_{l=1}^{W} x_{ij}^{(l)} \le 1, \forall\ v_i \in V \textit{ and } v_i \ne t \qquad (5)$$

$$\sum_{j=1}^{n}\sum_{l=1}^{W} x_{ji}^{(l)} \le 1, \forall\ v_i \in V \textit{ and } v_i \ne s \qquad (6)$$

$$\sum_{j=1}^{n}\sum_{l=1}^{W} x_{ij}^{(l)} - \sum_{j=1}^{n}\sum_{l=1}^{W} x_{ji}^{(l)} = 0,$$
$$\forall\ v_i \in V \textit{ and } v_i \ne s \textit{ and } v_i \ne t \qquad (7)$$

$$\sum_{j=1}^{n}\sum_{l=1}^{W} x_{sj}^{(l)} = 1 \qquad (8)$$

$$\sum_{j=1}^{n}\sum_{l=1}^{W} x_{jt}^{(l)} = 1 \qquad (9)$$

$$\sum_{l=1}^{W} y_{ij}^{(l)} \le 1, \forall\ \langle v_i, v_j\rangle \in E \qquad (10)$$

$$\sum_{j=1}^{n}\sum_{l=1}^{W} y_{ij}^{(l)} \le 1, \forall\ v_i \in V \textit{ and } v_i \ne t \qquad (11)$$

$$\sum_{j=1}^{n}\sum_{l=1}^{W} y_{ji}^{(l)} \le 1, \forall\ v_i \in V \textit{ and } v_i \ne s \qquad (12)$$

$$\sum_{j=1}^{n}\sum_{l=1}^{W} y_{ij}^{(l)} - \sum_{j=1}^{n}\sum_{l=1}^{W} y_{ji}^{(l)} = 0,$$
$$\forall\ v_i \in V \textit{ and } v_i \ne s \textit{ and } v_i \ne t \qquad (13)$$

$$\sum_{j=1}^{n}\sum_{l=1}^{W} y_{sj}^{(l)} = 1 \qquad (14)$$

$$\sum_{j=1}^{n}\sum_{l=1}^{W} y_{jt}^{(l)} = 1 \qquad (15)$$

$$\sum_{l=1}^{W} (x_{ij}^{(l)} + y_{ij}^{(l)}) \le 1, \forall\ \langle v_i, v_j\rangle \in E \qquad (16)$$

$$z_{ijk} = \sum_{l_1=1}^{W} \sum_{l_2=1}^{W} (x_{ij}^{(l_1)} + x_{jk}^{(l_2)} - 1) c_{v_j}(\lambda_{l_1}, \lambda_{l_2}),$$
$$\forall \langle v_i, v_j \rangle \text{ and } \langle v_j, v_k \rangle \in E \quad (17)$$

$$t_{ijk} = \sum_{l_1=1}^{W} \sum_{l_2=1}^{W} (y_{ij}^{(l_1)} + y_{jk}^{(l_2)} - 1) c_{v_j}(\lambda_{l_1}, \lambda_{l_2}),$$
$$\forall \langle v_i, v_j \rangle \text{ and } \langle v_j, v_k \rangle \in E \quad (18)$$

$$x_{ij}^{(l)}, y_{ij}^{(l)} \in \{0, 1\}, \forall \lambda_l \in \Lambda \text{ and } \langle v_i, v_j \rangle \in E \quad (19)$$

$$z_{ijk} \geq 0, \forall v_i, v_j, v_k \in V \quad (20)$$

$$t_{ijk} \geq 0, \forall v_i, v_j, v_k \in V \quad (21)$$

Ineq. (4) guarantees that only a wavelength is assigned for each physical link on the primary path. Ineqs (5) and (6) indicate that for each node at most one physical link is chosen as incoming to and outgoing from the node. Eq. (7) guarantees that each node on the primary path does not keep any fraction of the flow except the source node s (Eq. (8)) and the destination node t (Eq. (9)). Ineqs (10) to (15) are similar to Ineqs (4) to (9) which are the constraints for the backup path. Ineq. (16) says that for a physical link, if it will be used, then it is either on the primary or on the backup paths but not on both of them. Ineqs (17) and (20) (Ineqs (18) and (21)) are the costs of wavelength conversion at intermediate nodes on the primary path (backup path).

If there is a solution for the above integer programming, then $x_{ij}^{(l)} = 1$ means that link $\langle v_i, v_j \rangle$ has been chosen as a link on the primary path and assigned wavelength λ_l. If both $x_{ij}^{(l_1)} = 1$ and $x_{jk}^{(l_2)} = 1$, this means that the wavelength conversion switch at v_j on the primary path is set from wavelength λ_{l_1} to λ_{l_2}. Similarly, the backup path can be found and assigned wavelengths on its links as well setting the switches of wavelength conversion at intermediate nodes on it.

3.2 The hardness of the problem

We now consider a special case of the optimal edge-disjoint semilightpath problem where no wavelength conversion is allowed either on the primary path or on the backup path. Even for this special case, the problem is NP-hard, which is proved below.

Lemma 1 *Given an WDM network $G(V, E, \Lambda)$, assume that there are two wavelengths λ_1 and λ_2 on the network. The problem is to find two edge-disjoint lightpaths from s to t such that one of them is assigned wavelength λ_1, the other is assigned wavelength λ_2, and the cost sum of the two paths is minimized. The decision version of the problem is NP-complete.*

Proof Consider a directed graph $G(V, E)$ with $s, t \in V$, for each link $e \in E$, a pair of weights is associated with it. In our case, each link is assigned one of the pairs of weights: $(0, 0)$, $(0, 1)$, and $(1, 0)$. The two minimum-cost edge-disjoint problem is to ask whether there are two edge-disjoint paths from s to t in G with the cost of one of the paths using the weight in the first component of the pair, and the cost of the other using the weight in the second component of the pair such that the cost sum of the two paths is zero, which has been shown to be NP-compete [12].

The two minimum-cost edge-disjoint problem can be reduced to the optimal edge-disjoint semilightpath problem without wavelength conversion as follows. For the given $G(V, E)$, if the weights associated with a link $e \in E$ is $(0, 0)$, this means both wavelengths λ_1 and λ_2 are available on e; if the weight is $(1, 0)$, this means wavelength λ_2 is available on e but λ_1 is not; otherwise, the weight must be $(0, 1)$, this indicates that wavelength λ_1 is available on e but λ_2 is not. Therefore, if there is a solution for the two minimum-cost edge-disjoint problem, then there is a solution for the optimal edge-disjoint semilightpath problem without wavelength conversion and one of the paths is assigned λ_1 and the other is assigned λ_2. While the two minimum-cost edge-disjoint problem is NP-complete, the optimal edge-disjoint semilightpath problem without wavelength conversion is NP hard too. □

3.3 An approximate algorithm

Despite that there is an exact solution for the optimal edge-disjoint semilightpath problem by solving the corresponding integer programming, such a solution, however, is time consuming and very expensive. Here we focus on finding a feasible rather than an exact solution for the problem quickly. The technique used is the graph theoretic technique. We further assume that (i) fully switching is allowed at each node which means any incoming wavelength can be switched to any outgoing wavelength and the switching cost at a node is identical; (ii) the cost of traversing a physical link using different wavelengths is identical, i.e., $w(e, \lambda_1) = w(e, \lambda_2)$ if any $\lambda_i \in \Lambda(e)$, $i = 1, 2$. The basic idea behind the proposed algorithm is to reduce the problem to a well solved optimization problem on an auxiliary, directed weighted graph.

3.3.1 Construction of the auxiliary graph

Given an optical network $G(V, E, \Lambda)$, a subgraph $G_s = (V_s, E_s, \Lambda_{avail})$ of G is called the *residual network* of G, where $V_s = V$, $E_s \subseteq E$, $\Lambda_{avail}(e) \neq \emptyset$ for all $e \in E_s$, and $\Lambda_{avail}(e)$ ($\subseteq \Lambda(e)$) is the wavelength set on e in which every wavelength has not being used by any existent routing paths at this moment. Unless specified, from now on we assume the residual network of G is $G(V, E, \Lambda_{avail})$. For each $v \in V$, let $E_{in}(v) = \{\langle u, v \rangle : u \in V\}$ be the set of incoming links at v and $\Lambda_{avail}(\langle u, v \rangle) \neq \emptyset$; let $E_{out}(v) = \{\langle v, u \rangle : u \in V\}$ be the set of outgoing links at v and $\Lambda_{avail}(\langle v, u \rangle) \neq \emptyset$ in the residual network.

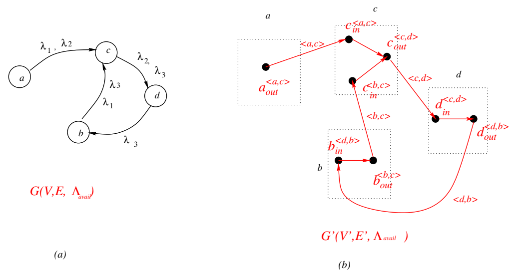

Figure 1: G and G' are the residual and auxiliary graphs.

The auxiliary graph $G' = (V', E', \omega)$ is constructed as follows. For each node $v \in V$, there are corresponding $|E_{in}(v)|$ incoming *edge-nodes* and $|E_{out}(v)|$ outgoing *edge-nodes* in V'. There is a directed link $\langle v^e_{in}, v^{e'}_{out}\rangle$ in E' if and only if there are at least a wavelength $\lambda \in \Lambda_{avail}(e)$ and another wavelength $\lambda' \in \Lambda_{avail}(e')$, and the wavelength conversation at v from λ to λ' is allowed, where $e \in E_{in}(v)$ and $e' \in E_{out}(v)$. The weight associated with link $\langle v^e_{in}, v^{e'}_{out}\rangle$ in G' is $\omega(\langle v^e_{in}, v^{e'}_{out}\rangle) = \sum_{\lambda_a \in \Lambda_{avail}(e)}^{\lambda_b \in \Lambda_{avail}(e')} c_v(\lambda_a, \lambda_b)/K_v$, where K_v is the number of different wavelength conversions at v. Note that the weight assigned to the link is the average cost of the costs of all possible conversions. For each link $e \in \langle u, v\rangle \in E$ in the residual network, there are two corresponding edge-nodes u^e_{out} and v^e_{in} in G' and u^e_{out} is derived from the link $e \in E_{out}(u)$ and v^e_{in} is derived from the link $e \in E_{in}(v)$. There is a directed link in G' from u^e_{out} to v^e_{in} and the weight associated with it is $\omega(\langle u^e_{out}, v^e_{in}\rangle) = \sum_{\lambda \in \Lambda_{avail}(e)} w(e, \lambda)/|\Lambda_{avail}(e)|$. Furthermore, there are two special nodes s' and t'' in V' which correspond to the source s and the destination t, respectively. There are directed links $\langle s', s^{e_1}_{out}\rangle$ and $\langle t^{e_2}_{in}, t''\rangle$ in G' for all $s^{e_1}_{out}, t^{e_2}_{in} \in V'$, where $s^{e_1}_{out}$ is derived from any link $e_1 = \langle s, u\rangle \in E_{out}(s)$ and $t^{e_2}_{in}$ is derived from any link $e_2 = \langle u, t\rangle \in E_{in}(t)$. Fig. 1 is such an example, where $G(V, E, \Lambda_{avail})$ in Fig. 1(a) is the residual network of an WDM optical network, and $G' = (V', E', \omega)$ in Fig. 1(b) is the corresponding auxiliary graph of G.

3.3.2 Approximate algorithm

Given the auxiliary graph $G'(V', E', \omega)$, apply Suurballe's algorithm [21] on G' to find two edge-disjoint paths P_1 and P_2 from s' to t'' such that $\sum_{e \in P_i}^{i=1,2} \omega(e)$ is minimized. The algorithm is described as follows.

`Find_Two_Paths` (G', s', t'', ω)

begin

$E_2 = \emptyset$;

for $i = 1$ **to** 2 **do**

  $E_{reserve} = \{\langle u, v\rangle \ : \langle v, u\rangle \in E_2\}$;

  find a shortest path P'_i from s' to t'' in

  $G'^{\,i} = (V', E' \cup E_{reserve} - E_2)$; /* $G'^{\,1} = (V', E', \omega)$ */

  $E_{intersect} = \{\langle u, v\rangle,\ \langle v, u\rangle\ :$

    $\langle u, v\rangle \in E_2\ \&\ \langle v, u\rangle \in E(P'_i)\}$;

  $E_2 = E_2 \cup E(P'_i) - E_{intersect}$

endfor

$G(V(E_2), E_2)$ is a subgraph of G' containing two edge-disjoint paths from s' to t'' if they exist.

end.

With the subgraph $G(V(E_2), E_2)$ of G', the two edge-disjoint paths in $G(V(E_2), E_2)$ from s' to t'' can be found easily, because each node except s' and t'' has incoming degree and outgoing degree 1. It has been shown that the weighted sum of the links in the two paths found by algorithm `Find_Two_Paths` is the minimum one [21]. It is easy to verify that if there are no such edge-disjoint paths in G' from s' to t'', then there are no two edge-disjoint paths in G from s to t too. But, for every semilightpath in G from s to t, there is a corresponding path in G' from s' to t''. Without loss of generality, we assume that there is a solution in G' consisting of the two paths P_1 and P_2. Then, there is a corresponding approximate solution consisting of P_{11} and P_{22} in G. In the following a better approximate solution than P_{11} and P_{22} in G consisting of P'_1 and P'_2 is described as follows.

Construct a subgraph $G_i(V_i, E_i, \Lambda_{avail})$ of $G(V, E, \Lambda_{avail})$ using P_i, where V_i is the set of endpoints of links $e \in E$ and the corresponding edge-node of e is on P_i, and E_i is the set of links in the residual network G corresponding to the edge-nodes in P_i. Apply the algorithm due to Liang and Shen [13] to find an optimal semilightpath P'_i in G_i from s to t, $i = 1, 2$. In the following we first show that the cost sum $C(P'_1) + C(P'_2)$ of P'_1 and P'_2 is better than the cost sum $C(P_{11}) + C(P_{22})$ of P_{11} and P_{22} in G. We then show that P'_1 and P'_2 form an approximate solution for the problem.

Lemma 2 *Let $C(P'_i)$ be the cost of P'_i in $G(V, E, \Lambda_{avail})$ and $\omega(P_i)$ $(= \sum_{e \in P_i} \omega(e))$ be the weighted sum of the links of P_i in $G'(V', E', \omega)$, $i = 1, 2$. Assume that P_{ii} is the corresponding semilightpath in G derived from P_i, $i = 1, 2$. Then, $C(P'_1) + C(P'_2) \le C(P_{11}) + C(P_{22})$. P'_1 and P'_2 in $G(V, E, \Lambda_{avail})$ are edge-disjoint.*

Proof Since P_{11} and P_{22} are the corresponding semilightpaths of P_i in G, then $C(P_{11}) + C(P_{22}) = \omega(P_1) + \omega(P_2)$ by the definition. Though P_{ii} is a semilightpath in G_i from s to t, P'_i is an optimal semilightpath in it from s to t. So, $C(P'_i) \le C(P_{ii})$, $i = 1, 2$. Thus, $C(P'_1) + C(P'_2) \le C(P_{11}) + C(P_{22})$.

Assume that P'_1 and P'_2 in G are not edge-disjoint. Let $e = \langle u, v\rangle$ be the first link shared by the both paths. By the definition of $G'(V', E', \omega)$, there are two edge-nodes u^e_{out} and v^e_{in} in G' and there is a link e^* in E' from u^e_{out} to v^e_{in}, where u^e_{out} is derived from $e \in E_{out}(u)$ and v^e_{in} is derived from $e \in E_{in}(v)$. Since $e^* \in E'$ and e^* is in both P_1 and P_2, then P_1 and P_2 in G' are not edge-disjoint. This contradicts the assumption that P_1 and P_2 in G' are edge-disjoint. Therefore, P'_1 and P'_2 in G are edge-disjoint. □

Theorem 1 *Given a residual optical network $G(V, E, \Lambda_{avail})$ and a connection request, there is an approximate algorithm for the problem. The algorithm takes $O(nd + nW^2 + m\log_2 n + nW\log_2(nW))$ time, where d is the maximum degree of nodes in G and W is the number of wavelengths in G.*

Proof The construction of the auxiliary graph G' takes $O(m + nd)$ time because G' contains $2m$ nodes and at most $O(m+nd)$ links, where d is the maximum degree of nodes in the residual network $G(V, E, \Lambda_{avail})$. Finding two edge-disjoint paths in G' from s' to t'' with minimizing the cost sum of the two paths takes $O(nd + m\log_2 n)$ time using an efficient algorithm for the single-source shortest path problem [7] and $m \leq nd$. Constructing a subgraph G_i of G derived from P_i takes $O(n)$ time because G_i contains $O(n)$ nodes and links and finding an optimal semilightpath G_i takes $O(nW^2 + nW\log_2(nW))$ time [?, 13], $i = 1, 2$. Thus, the algorithm takes $O(nd + nW^2 + m\log_2 n + nW\log_2(nW))$ time. □

Finally we analyze the performance ratio between the approximate solution obtained and the exact solution. We have the following theorem.

Theorem 2 *Given a residual optical network $G(V, E, \Lambda_{avail})$ and a connection request, the solution delivered by the approximate algorithm is twice of the optimal, assuming that the cost of wavelength conversion at a node is no greater than the cost of traversing any its incident link.*

Proof Let P' be a semilightpath in the residual network G. The cost $C(P')$ of P' is $C(P') = C_w(P') + C_c(P')$, where $C_w(P')$ is the sum of the costs of traversing the links, and $C_c(P')$ is the sum of the costs of wavelength conversions on P'. Then,

$$C_c(P') \quad \leq \quad C_w(P') \tag{22}$$

by the assumption that the cost of wavelength conversion at a node is no greater than the cost of traversing any its incident link.

Assume that $A^{'*}$ consisting of two edge-disjoint semilightpaths $P_1^{'*}$, and $P_2^{'*}$ is an exact solution of the problem and A' consisting of two edge-disjoint semilightpaths P_1' and P_2' is an approximate solution delivered by the approximate algorithm, where P_1' and P_2' are obtained from the subgraphs G_1 and G_2 of G derived from the two edge-disjoint paths P_1 and P_2 in the auxiliary graph $G'(V', E', \omega)$. Note that the cost sum $\omega(P_1) + \omega(P_2)$ of P_1 and P_2 in G' is the minimum one among all the two edge-disjoint paths from s' to t''. By Lemma 2, we have $C(P_1') + C(P_2') \leq \omega(P_1) + \omega(P_2)$. For the given $P_1^{'*}$ and $P_2^{'*}$ in G, let P_1^* and P_2^* be the two corresponding edge-disjoint paths in G'. Then, by the approximate algorithm we have $\omega(P_1) + \omega(P_2) \leq \omega(P_1^*) + \omega(P_2^*)$ because the cost sum of P_1 and P_2 in G' is the minimum one.

Let P be a directed path in G' from s' to t''. The weighted sum $\omega(P)$ of the links on P can be rewritten as follows.

$$\omega(P) \quad = \quad W_e(P) + W_v(P), \tag{23}$$

where $W_e(P)$ is the weighted sum of links on P corresponding to traversing links in G using wavelengths and $W_v(P)$ is the weighted sum of links of P' corresponding to the wavelength conversions at intermediate nodes on a semilightpath.

By the initial assumption (i) the cost of wavelength conversion at a node is identical. Let C_u be the cost of wavelength conversion at a node u. Because there is a corresponding semilightpath P' in G for each directed path P in G', and the weight associated with every link in P corresponding to wavelength conversion at an intermediate node on P' is the average cost of all conversion costs at the node. Thus, we have $C_w(P') \geq C_v(P')$ by Eq. (22). While $\omega(P) = W_e(P) + W_v(P) = C_w(P') + W_v(P)$ and $W_v(P) = \sum_{u \in S} c_u(\lambda_a, \lambda_b)/K_u$, where $\lambda_a \in \Lambda_{avail}(e)$ and $e \in E_{in}(u)$, $\lambda_a \in \Lambda_{avail}(e')$ and $e' \in E_{out}(u)$, S is the set of the intermediate nodes, and K_u is the number of possible wavelength conversions at u. Let n_u be the number of common wavelengths shared on links e and e', then $n_u = |\Lambda_{avail}(e) \cap \Lambda_{avail}(e')|$. Therefore, we have $W_v(P) = \sum_{u \in S}(1 - \frac{|\Lambda_{avail}(e) \cap \Lambda_{avail}(e')|) \times C_u}{|\Lambda_{avail}(e)| \times |\Lambda_{avail}(e')|})$ $\leq \sum_{u \in S} C_u = C_c(P'')$, where P'' is a semilightpath which consists of the same optical links as P', the only difference between P' and P'' is the wavelength assignment. We knew that $C_w(P'') \geq C_c(P'') = W_v(P)$, and $C_w(P'') = C_w(P')$ according to the initial assumption (ii) the cost of traversing a link with different wavelengths is identical. Therefore, $W_e(P) = C_w(P') = C_w(P'') \geq C_c(P'') = W_v(P)$. Then, $\frac{W_e(P)}{W_v(P)} \geq 1$. Thus, we have $\omega(P_1^*) + \omega(P_2^*) = W_e(P_1^*) + W_e(P_2^*) + W_v(P_1^*) + W_v(P_2^*)$ and $C(P_1^{'*}) + C(P_2^{'*}) = C_w(P_1^{'*}) + C_w(P_2^{'*}) + C_c(P_1^{'*}) + C_c(P_2^{'*})$. While $W_e(P_1^*) + W_e(P_2^*) = C_w(P'*_1) + C_w(P'*_2)$ by assumption (ii), we have $W_e(P_1^*) + W_e(P_2^*) = C_w(P_1^{'*}) + C_w(P_2^{'*})$. Thus, the performance ratio between the approximate and the exact solution is as follows.

$$\frac{C(P_1') + C(P_2')}{C(P_1^{'*}) + C(P_2^{'*})} \leq \frac{\omega(P_1) + \omega(P_2)}{C(P_1^{'*}) + C(P_2^{'*})} \leq \frac{\omega(P_1^*) + \omega(P_2^*)}{C(P_1^{'*}) + C(P_2^{'*})}$$
$$= \frac{W_e(P_1^*) + W_e(P_2^*) + W_v(P_1^*) + W_v(P_2^*)}{C(P_1^{'*}) + C(P_2^{'*})}$$
$$= \frac{W_e(P_1^*) + W_e(P_2^*) + W_v(P_1^*) + W_v(P_2^*)}{C_w(P_1^{'*}) + C_w(P_2^{'*}) + C_c(P_1^{'*}) + C_c(P_2^{'*})}$$
$$= \frac{C_w(P_1^{'*}) + C_w(P_2^{'*}) + W_v(P_1^*) + W_v(P_2^*)}{C_w(P_1^{'*}) + C_w(P_2^{'*}) + C_c(P_1^{'*}) + C_c(P_2^{'*})}$$
$$= 1 + \frac{W_v(P_1^*) + W_v(P_2^*) - (C_c(P_1^{'*}) + C_c(P_2^{'*}))}{C_w(P_1^{'*}) + C_w(P_2^{'*}) + C_c(P_1^{'*}) + C_c(P_2^{'*})}$$
$$\leq 1 + \frac{W_v(P_1^*) + W_v(P_2^*)}{C_w(P_1^{'*}) + C_w(P_2^{'*}) + C_c(P_1^{'*}) + C_c(P_2^{'*})}$$
$$\leq 1 + \frac{W_v(P_1^*) + W_v(P_2^*)}{C_w(P_1^{'*}) + C_w(P_2^{'*})} \leq 1 + \frac{W_v(P_1^*) + W_v(P_2^*)}{W_e(P_1^*) + W_e(P_2^*)}$$
$$\leq 1 + \frac{W_e(P_1^*) + W_e(P_2^*)}{W_e(P_1^*) + W_e(P_2^*)} = 2. \ \square$$

4 Minimizing Both the Load and the Routing Cost

In this section an approach is proposed to reduce the number of reconfigurations if extra care is taken during establishing routing paths for user connection requests. That is, during finding two edge-disjoint semilightpaths for a connection request, the network load is also taken into consideration. As results, the performance of the entire network is improved dramatically through the reduction of the number of reconfigurations. The problem therefore is to find two edge-disjoint semilightpaths in G from s to t such that not only the cost sum of the two paths is minimized but also the network load is minimized. However, considering either one of the optimization objectives is NP-hard, not to mention the both optimization objectives being considered simultaneously. Accordingly, a simpler version in which the network load is considered only is studied first, followed by extending the techniques for the simpler version to solve the problem.

4.1 Minimizing the network load

For a given connection request from s to t, a simpler version of the problem is to find two edge-disjoint semilightpaths from s to t such that the network load ρ is minimized. To do so, we first reduce the problem to an optimization problem on an auxiliary graph. We then solve the optimization problem on the graph, which gives an approximate solution for the problem. In particular, we use an exponential function to assign weights to the links in the auxiliary graph. The routes are then chosen using the weights of links in the paths.

Given a network load ϑ with $0 < \vartheta < 1$, the auxiliary graph $G_c = (V_c, E_c, \omega_c, \vartheta)$ from the residual network $G(V, E, \Lambda_{avail})$ is constructed as follows. For each node $v \in V$, if a link $e \in E_{in}(v)$ and $0 \leq U(e)/N(e) < \vartheta$, then, there is a corresponding *edge-node* v_{in}^e in V_c for e. Similarly, if a link $e' \in E_{out}(v)$ and $0 \leq U(e')/N(e') < \vartheta$, then, there is a corresponding edge-node $v_{out}^{e'}$ in V_c for e'. There is a directed link $\langle v_{in}^e, v_{out}^{e'} \rangle$ in E_c if and only if there are a wavelength on e and another wavelength on e' and the wavelength conversation at v is allowed. The weight associated with the link $\omega_c(\langle v_{in}^e, v_{out}^{e'} \rangle)$ in G_c is 0. For every link $e = \langle u, v \rangle \in E$, there are two edge-nodes u_{out}^e and v_{in}^e in G_c, where u_{out}^e is generated due to $e \in E_{out}(u)$ and v_{in}^e is generated due to $e \in E_{in}(v)$. There is a directed link from u_{out}^e to v_{in}^e in G_c with weight $\omega_c(u_{out}^e, v_{in}^e) = a^{\frac{U(e)+1}{N(e)}} - a^{\frac{U(e)}{N(e)}}$, where $a > 1$ is constant. Note that the weight assigned to a link in the above expression is an exponential function of its load. This heuristic function helps to choose the links with light loads rather than heavy loads as the links for the routing path. Moreover, there are two special vertices $s', t'' \in V_c$. For every edge-node corresponding to an outgoing link from s, there is a directed link in E_c from s' to the node with weight 0. Similarly, for every edge-node corresponding to an incoming link at t, there is a directed link in E_c from the node to t'' with weight 0. Note that the construction of G_c is similar to that of G'. The only difference is the weight assignments of links, and some links in G' are not appeared in G_c due to that those links loads are beyond the threshold ϑ. Therefore, G_c is a subgraph of G'.

Having a weighted directed graph G_c, apply Suurballe's algorithm to find two edge-disjoint paths from s' to t'' in G_c such that the weighted sum of the two paths is minimized. If there is no feasible solution for the graph G_c, this indicates that the chosen network load ϑ is too small, we need to increase ϑ, reconstruct G_c and run the algorithm again until the two paths can be found from the current G_c or $\vartheta > \max_{e \in E}\{\frac{U(e)+1}{N(e)}\}$. If $\vartheta = \max_{e \in E}\{\frac{U(e)+1}{N(e)}\}$ and there is still no solution for the connection request, the request is dropped, which means that no such paths exist. Without loss of generality, we assume there is a solution and P_1 and P_2 are the two found paths in G_c. For each P_i, a subgraph G_i of G then can be induced, using the approach in the previous section. Find an optimal semilightpath P_i' from G_i, $i = 1, 2$. Clearly, P_1' and P_2' are edge-disjoint in G too. However, in practice the network load ϑ is not known in advance, therefore, a suitable ϑ is needed, but we know $\vartheta_{min} \leq \vartheta \leq \vartheta_{max}$, where $\vartheta_{min} = \min_{e \in E}\{\frac{U(e)+1}{N(e)}\}$ and $\vartheta_{max} = \max_{e \in E}\{\frac{U(e)+1}{N(e)}\}$. Thus, the following algorithm is to find an approximate solution for the two edge-disjoint semilightpaths in G from s to t with minimizing the network load ϑ.

```
Find_Two_Paths_Mincog (G, ϑ)
begin
    finish=false; ϑ = ϑ_min; Δ = ϑ_max − ϑ_min;
    j_0 = −⌈log_2(ϑ_max − ϑ_min)⌉; j = j_0;
    repeat
        Construct G_c(V_c, E_c, ω_c, ϑ);
        Find P_1 and P_2 in G_c from s' to t'',
        by applying Suurballe's algorithm to G_c.
        if    no such two paths exist
              then   j = j − 1; ϑ = ϑ + Δ/2^j;
              else   finish= true
    until (finish) or (j < 0);
    return (ϑ, P_1, P_2);
end.
```

Theorem 3 *Given a residual optical network $G(V, E, \Lambda_{avail})$ and a connection request, there is an approximate algorithm delivering an approximate solution which is 3 times of the optimal. The algorithm takes $O(nd \log_2 \frac{1}{\Delta} + m \log_2 n \log \frac{1}{\Delta} + nW^2 + nW \log_2(NW))$ time, where d is the maximum degree of nodes in G, $\Delta = \vartheta_{\max} - \vartheta_{\min}$, and W is the number of wavelengths in G.*

Proof Let ϑ^* be the exact solution and ϑ an approximate solution of the simpler version. For a given connection request, assume that the solution exists. Then,

there is a k ($j_0 \geq k$) such that $\vartheta = \vartheta_{min} + \sum_{j=j_0}^{k} \Delta/2^j$. By the approximate algorithm, we have $\vartheta^* > \vartheta_{min} + \sum_{j'=j_0}^{k+1} \Delta/2^{j'}$. Thus, the performance ratio between the approximate and the exact solutions is
$\frac{\vartheta}{\vartheta^*} = \frac{\vartheta_{min}+\sum_{j=j_0}^{k} \Delta/2^j}{\vartheta_{min}+\sum_{j'=j_0}^{k+1} \Delta/2^{j'}} = 1 + \frac{\Delta/2^k}{\vartheta_{min}+\sum_{j'=j_0}^{k+1} \Delta/2^{j'}}$
$\leq 1 + \frac{\Delta/2^k}{\Delta/2^{k+1}} < 3$.

We now analyze the algorithmic complexity. The construction of G_c takes $O(m+nd)$ time because G_c contains $2m$ nodes and at most $O(nd+m)$ links. Finding an approximate solution ϑ takes $O((m + nd + m\log_2 n)\log_2 \frac{1}{\Delta})$. Then, finding two edge-disjoint semilightpaths from the induced subgraph of G takes $O(nW^2 + nW\log_2(NW))$ time. Thus, the algorithm takes $O(nd\log_2 \frac{1}{\Delta} + m\log_2 n \log \frac{1}{\Delta} + nW^2 + nW\log_2(NW))$ time. □

4.2 Optimizing network load and routing cost

We extend the techniques for the simpler version to solve the problem by presenting an approximate algorithm for it. This approximate solution consists of two phases. In the first phase, we minimize the network load by applying algorithm `Find_Two_Paths_MinCog` which will return a feasible network load, ϑ. In the second phase an auxiliary weighted, directed graph $G_{rc} = (V_c, E_c, \omega, \vartheta)$ is constructed, which is exactly the same as G_c except that the weight assignment of the links in G_{rc} is totally different from that in G_c. In this latter case the routing cost is also taken into account. The weight assignment of the links in G_{rc} is described as follows.

Let u^e_{out} and v^e_{in} be two edge-nodes in G_{rc} derived from link $e = \langle u, v\rangle \in E$, where u^e_{out} is generated due to $e \in E_{out}(u)$, v^e_{in} is generated due to $e \in E_{in}(v)$, and $U(e)/N(e) < \vartheta$. The weight associated with the link from u^e_{out} to v^e_{in} in G_{rc} is $\omega(\langle u^e_{out}, v^e_{in}\rangle) = \sum_{\lambda\in\Lambda_{avail}(e)} w(e,\lambda)/N(e)$, which is the average of all possible weights on link e using different wavelengths.

Let e and e' be the two links in E sharing a common node v with both $U(e)/N(e) < \vartheta$ and $U(e')/N(e') < \vartheta$. Assume the head $head(e) = v$ of e and $tail(e') = v$ of e'. Then, there is a directed link in G_{rc} from an edge-node v^e_{in} to another edge-node $v^{e'}_{out}$ with weight $\omega(\langle v^e_{in}, v^{e'}_{out}\rangle) = \sum_{\lambda_a\in\Lambda_{avail}(e)}^{\lambda_b\in\Lambda_{avail}(e')} c_v(\lambda_a,\lambda_b)/K_v$, where K_v is the number of different wavelength conversions at node v (=$head(e)$). Note that the weight assigned to the link corresponding the wavelength conversion at v is an average of the all possible wavelength conversions at v. Moreover, there are two special nodes s' and t'' in V_c. There are directed links $\langle s', s^{e_1}_{out}\rangle$ and $\langle t^{e_2}_{in}, t''\rangle$ in G_{rc} for all $e_1 \in E_{out}(s)$ with $U(e_1)/N(e_1) < \vartheta$; and all $e_2 \in E_{in}(t)$ with $U(e_2)/N(e_2) < \vartheta$. All of those links are assigned weight zeros.

Apply Suurballe's algorithm for finding two edge-disjoint paths in G_{rc} from s' to t''. Then, two edge-disjoint semilightpaths in the subgraphs of G from s to t, induced by the nodes and links derived from the two found paths in G_{rc}, can be found easily. Thus, an approximate solution for the problem is obtained.

References

[1] A. S. Acampora. Partially reconfigurable multihop lightwave networks. *GLOBECOM'90*, Dec., 1990.

[2] A. Aggarwal et al. Efficient routing in optical networks. *J. of the ACM* **46:**973–1001, 1996.

[3] M. Alanyali and E. Ayanoglu. Provisioning algorithms for WDM optical networks. *Infocom'98*, 1998.

[4] J. Armitage et al. Design of a survivable WDM photonic network. *Infocom'97*, IEEE, 1997.

[5] I. Chlamtac et al. Lightpath (wavelength) routing in large WDM networks. *IEEE Journal on Selected Areas in Communications* **14:**909–913, 1996.

[6] I. Chlamtac et al. Lightpath communications: a novel approach to high bandwidth optical WDN's. *IEEE Trans. Commun.* **40:**, 1992.

[7] M. L. Fredman and R. E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *J. ACM*, **34:**596–615, 1987.

[8] B. Gavish and I. Neuman. Routing in a network with unreliable components. *IEEE Trans Commun.*, **40:**1248–1257, 1992.

[9] P. E. Green. *Fiber- Optic Communication Networks.* Prentice Hall, 1992.

[10] E. Karasan and E. Ayanoglu. Performance of WDM transport networks *IEEE J. Selected Area in Commun.*, **16:**1081–1096, 1998.

[11] M. Kodialam and T. V. Lakshman. Dynamic routing of bandwidth guaranteed tunnels with restoration. *Infocom'00*, 2000.

[12] C-L Li et al. Finding disjoint paths with different path costs: complexity and algorithms. *Networks*, **22:**653–667, 1992.

[13] W. Liang and X. Shen. Improved lightpath (wavelength) routing in large WDM networks. *IEEE Trans. Commun.*,**48:**1571–1479, Sept., 2000.

[14] A. D. McAulay. *Optical Computer Architectures.* John Wiley, 1991.

[15] G. Mohan and A. K. Somani. Routing dependable connections with specified failure restoration guarantees in WDM networks. *Infocom'00*, 2000.

[16] A. Mokhtar and M. Azizoglu. Adaptive wavelength routing in all-optical networks. *IEEE/ACM Trans. Networking* **6:**197–206, 1998.

[17] N. Nagatsu et al. Optical path cross-connect system scale evaluation using path accommodation design for restricted wavelength multiplexing. *IEEE J. Selected Area Commun.*, **14:**893–902, 1996.

[18] A. Narula-Tam and E. Modiano. Dynamic load balancing for WDM-based packet networks. *Infocom'00*, 2000.

[19] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity.* Prentice-Hall, 1982.

[20] R. Ramaswami. Multi-wavelength lightwave networks for computer communication. *IEEE Communications Magazine* **31:**78–88, 1993.

[21] J. W. Suurballe. Disjoint paths in a network. *Networks* **4:**125–145, 1974.

[22] R. J. Vitter and D. H. C. Du. Distributed computing with high-speed optical networks. *IEEE Computer* **26:**8–18, 1993.

[23] T. -H. Wu. *Fiber Network Service Survivability.* Artech House, 1992.