New and Improved Approaches for Shared-Path Protection in WDM Mesh Networks

Canhui (Sam) Ou, *Student Member, IEEE*, Jing Zhang, *Student Member, IEEE*, Hui Zang, *Member, IEEE*, Laxman H. Sahasrabuddhe, *Member, IEEE*, and Biswanath Mukherjee, *Member, IEEE*

***Abstract*—This paper investigates the problem of dynamic survivable lightpath provisioning in optical mesh networks employing wavelength-division multiplexing (WDM). In particular, we focus on shared-path protection because it is resource efficient due to the fact that backup paths can share wavelength links when their corresponding working paths are mutually diverse. Our main contributions are as follows. 1) First, we prove that the problem of finding an eligible pair of working and backup paths for a new lightpath request requiring shared-path protection under the current network state is NP-complete. 2) Then, we develop a heuristic, called CAFES, to compute a feasible solution with high probability. 3) Finally, we design another heuristic, called OPT, to optimize resource consumption for a given solution. The merits of our approaches are that they capture the essence of shared-path protection and approach to optimal solutions without enumerating paths. We evaluate the effectiveness of our heuristics and the results are found to be promising.**

***Index Terms*—Fault management, lightpath, optical network, provisioning, shared-path protection, wavelength-division multiplexing (WDM).**

I. Introduction

In a wavelength-routed optical network, the failure of a network element (e.g., fiber, crossconnect, etc.) can cause the failure of several lightpaths, thereby leading to large data and revenue loss. Protection [19], a proactive procedure in which spare capacity is reserved during lightpath setup, can be employed to combat such failures. Protection schemes can be classified by the type of routing used (link-based versus path-based) and by the type of resource sharing (dedicated versus shared) [5], [14]. A path that carries traffic during normal operation is known as a *working* path.[1] When a working path fails, the lightpath is rerouted over a *backup* path.

We consider the problem of dynamic survivable lightpath provisioning against single-fiber failures.[2] Specifically, we focus on shared-path protection because of its desirable resource efficiency resulting from backup sharing. Protection approaches to optimizing resource utilization for a given traffic matrix [4], [15], [19], [23] do not apply because lightpath requests come and go in the dynamic provisioning case. Under such a scenario, a network management system needs to compute two link-disjoint paths—a dedicated working path and a shared backup path—for an incoming lightpath request based on the current network state. We concentrate on computing link-disjoint paths for each incoming lightpath request with the assumptions that existing lightpaths cannot be disturbed and no knowledge of future arrivals is available at the time of provisioning this lightpath request. While we consider full wavelength-convertible networks here, the extension to the wavelength-continuous case is straightforward.

Much work has been conducted on dynamic survivable lightpath protection in an optical WDM network and on dynamic routing of restorable bandwidth-guaranteed connections in a multiprotocol label switching (MPLS) network. Although some papers are devoted to the MPLS context, their basic ideas—with appropriate variations, e.g., quantized bandwidth granularities—are applicable to the shared-path protection problem in a WDM mesh network with full wavelength conversion at each node. Table I provides an overview of some related work, more elaboration on which follows. In what follows, the term "lightpath" will be used in WDM context while the term "connection" will be used in MPLS context.

The most desirable property of shared-path protection is its resource efficiency resulting from backup sharing. Consequently, how to increase backup sharing based on different cost models is of particular interest and has been reported in [2], [7], [10], [12], [13], [20], and [21]. Since backup sharing depends on the routes of working paths, most of existing work computes a backup path after the working path is determined. A natural questions is: Can we jointly compute a working path and a backup path for a lightpath request in polynomial time in a way similar to Suurballe's algorithm?[3] We show that the answer is no by proving that the problem of finding an eligible pair of working and backup paths under shared-path-protection constraints for a lightpath request with respect to existing lightpaths is NP-complete. Then, we develop a heuristic, called CAFES, to compute a feasible solution (i.e., two link-disjoint paths) with high probability. A drawback of computing a backup path after fixing the working path is that the working and backup paths combined may use more resources than

Manuscript received June 20, 2003; revised December 22, 2003. This work was supported by the NSF under Grant ANI-98-05285 and Sprint Advanced Technology Labs. A short, summarized version of this work was presented at the IEEE International Conference on Communications (ICC '03) in May 2003.

C. Ou, J. Zhang, and B. Mukherjee are with the Department of Computer Science, University of California, Davis, CA 95616 USA (e-mail: ouc@cs.ucdavis.edu; zhangj@cs.ucdavis.edu; mukherje@cs.ucdavis.edu).

H. Zang is with the Sprint Advanced Technology Labs, Burlingame, CA 94010 USA (e-mail: hzang@sprintlabs.com).

L. H. Sahasrabuddhe is with the SBC Services, Inc., San Ramon, CA 94583 USA (e-mail: ls9526@sbc.com).

Digital Object Identifier 10.1109/JLT.2004.825346

[1]Working path is also referred to as primary path, active path, and service path in the literature.

[2]We focus on single-fiber failures because they are the predominant form of failures in communication networks.

[3]Suurballe's algorithm [22] jointly computes two link-disjoint paths of minimal total cost without consideration of backup sharing.

TABLE I
COMPARISON OF RELATED WORK ON DYNAMIC SHARED-PATH-PROTECTED LIGHTPATH/CONNECTION PROVISIONING ON WDM/MPLS MESH NETWORKS

Research Work	Objective	Path Link	Centralized Distributed	Info.	Wavelength Conversion	Deterministic Probabilistic	Contributions (in brief)
Bouillet et al. [1], [2]	Minimize the total cost of working and backup paths for each lightpath	P	C	A	Y	P & D	Stochastic approaches; cost model; K-shortest path routing.
Elie-Dit-Cosaque et al. [7]		P	D	F, P	Y	D	Protection-sharing table.
Mohan et al. [16]		P	C	A	N	D	Primary-backup sharing; Cost model for route computation.
Ramamurthy et al. [18]		P	D	F	Y	D	Performance comparison of different schemes.
Su et al. [20], [21]		L	D	F	Y	D	Bucket-based link metric; ILP & two-step heuristic.
Xin et al. [24]		P	C	F	Y	D	K-shortest path routing.
Xiong et al. [25]		P	C & D	F & P	Y	D	ILP formulations.
Our work		P	C	A	Y	D	NP-complete proof; heuristic for optimization; heuristic for finding a feasible solution.
Kodialam et al. [10], [11]	Minimize the total cost of working and backup paths for each connection	P & L	C & D	F, N, P	Y	D	ILPs for different scenarios & a heuristic based on primal-dual and LP-relaxation.
Li et al. [12]		P	C	F	Y	D	Two-step heuristic using a bucket-like link metric; distributed signaling.
Liu et al. [13]		P	D	A	Y	D	Aggregating per-flow information with a matrix; successively updating existing backups.
Qiao et al. [17]		P	D	P	Y	D	ILP & two-step heuristic; distributed signaling.

Path/Link: Path protection or link protection.
Centralized/Distributed: Whether the algorithm is centralized or distributed.
Info.: The amount of information needed. A: aggregated lightpath/connection information; F: full per-lightpath/connection information; N: no information about existing lightpaths/connections; P: partial information about existing lightpaths/connections.
Wavelength conversion: Whether the work applies to wavelength-continous network or wavelength-convertible network.
Please note that the basic ideas of the work in [10], [11], [12], [13], [17] (which is devoted to MPLS networks) are applicable to WDM networks.

necessary. Therefore, we provide a new heuristic, called OPT, to iteratively optimize the resource consumption for a given solution. While our focus is on dynamic lightpath provisioning, the two heuristics can be readily applied to static lightpath provisioning in which the traffic matrix is known *a priori*. For example, we can apply CAFES to find a feasible solution for every lightpath request in the traffic matrix in the first phase; and in the second phase, we can apply OPT to every solution found in the first phase to optimize the overall resource consumption.

While it is desirable to have complete information about the routing and wavelength assignment of the existing lightpaths to decide backup sharing, complete information may not always be available due to control and management concerns. Three scenarios—complete, partial, and no information—have been introduced in [10] and further examined in [11], [17] to quantify the impact of the amount of available information on backup sharing. The amount of information in the complete-information scenario can be reduced without sacrificing backup sharing by aggregating backup-sharing information via various techniques as shown in [7], [13], [16], [20], [21]. Our study utilizes aggregated information, which will be discussed in Section II-A.

One possible limitation of shared-path protection is that backup paths may sometimes become longer due to backup sharing [18]. The relation between backup sharing and backup-path hop distance have been show to be that one trades off another in [2], [25]. Our study prevents backup paths from being unnecessarily detoured while encouraging backup sharing in a way related to the approach in [25].

While the dynamic shared-path-protection problem can be formulated as an integer linear program (ILP) [17], [21], [25], ILPs are not scalable based on current computational power and may not be suitable for online computation. As reported in [25], the average processing time per connection request is 0.5 s for a 15-node network with 200 requests (on a PC with a 1.5 GHz CPU); the average processing time per connection request quickly increases to 8.5 s for a 46-node network with 200 requests. Therefore, we resort to efficient heuristics in our current study.

The rest of the paper is organized as follows. Section II formally states the problem and analyzes its complexity. Section III presents the CAFES heuristic for finding a feasible solution. Section IV presents another heuristic, OPT, for optimizing a given solution. Section V evaluates the performance of our heuristics via simulations. Section VI concludes this paper.

II. PROBLEM STATEMENT AND COMPLEXITY ANALYSIS

A. Problem Statement

We first define the notations and then formally state the dynamic shared-path-protected lightpath-provisioning problem. A network is represented as a weighted, directed graph $G = (V, E, C, \lambda)$, where V is the set of nodes, E is the set of unidirectional fibers (referred to as links), $C : E \rightarrow R^+$ is the cost function for each link (where R^+ denotes the set of positive real numbers), and $\lambda : E \rightarrow Z^+$ specifies the number of wavelengths on each link (where Z^+ denotes the set of positive integers).

We use λ_f^e to denote the number of free wavelengths on link $e \in E$. We denote the set of existing lightpaths by $\mathcal{L} = \{(l_w^i, l_b^i, t_a^i, t_h^i)\}$, where the quadruple $(l_w^i, l_b^i, t_a^i, t_h^i)$ specifies the working path, the backup path, the arrival time, and the holding time, in order, for the i^{th} lightpath. We denote the current lightpath request by (l_w, l_b, t_a, t_h). We represent the cost of l_w and l_b using $C_w(l_w)$ and $C_b(l_w, l_b)$, respectively.

We associate a conflict set with a link[4] to identify the sharing potential between backup paths. The conflict set ν_e for link e defines the set of links used by those working paths whose backup paths utilize wavelengths on link e. The conflict set ν_e for link e can be represented as an integer set, $\{\nu_e^{e'} | \forall e' \in E, 0 \leq \nu_e^{e'} \leq \lambda(e')\}$, where $\nu_e^{e'}$ specifies the number of working paths that traverse link e' and are protected by link e (and their corresponding backup paths traverse link e). The number of wavelengths reserved for backup paths on link e is thus $\nu_e^* = \max_{\forall e'} \{\nu_e^{e'}\}$. Clearly, the union of the conflict sets for all the links aggregates the per-lightpath-based information, and the size of the conflict set depends only on the number of links, not on the number of lightpaths. In the absence of such a mechanism as conflict set, per-lightpath-based information is necessary for identifying shareable backup channels [1]. It is, thus, advantageous to use conflict set since the number of lightpaths can be significantly more than the number of links.

The working and backup paths l_w and l_b satisfy the *shared-path-protection constraints* with respect to the existing lightpaths as follows:

- C.1 l_w and l_b are link disjoint.[5]
- C.2 l_w and l_w^i, $1 \leq i \leq |\mathcal{L}|$, do not utilize the same wavelength on any common link they traverse.
- C.3 l_w does not share any wavelength with l_b^i, $1 \leq i \leq |\mathcal{L}|$, on any common link they traverse.
- C.4 l_b and l_b^i can share a wavelength on a common link only if l_w and l_w^i are link disjoint.

We now formally state the dynamic shared-path-protected lightpath-provisioning problem as follows: Given a WDM network as $G = (V, E, C, \lambda)$ and the set of existing lightpaths (or the associated conflict sets $\{\nu_e | e \in E\}$), route each incoming lightpath request under shared-path-protection constraints while minimizing the total cost of the working and backup paths. In the following subsection, we show that the existence version of this problem is NP-complete.

B. Complexity Analysis

We formally state the decision version of the dynamic shared-path-protected lightpath-provisioning (DSPLP) problem below and prove that it is NP-complete.

Instance: A graph $G = (V, E, C, \lambda)$, the set of existing lightpaths $\mathcal{L}$ (or the set of conflict sets $\{\nu_e | e \in E\}$), and a lightpath request from s to d $(s, d \in V)$.

Question: Do there exist from s to d two paths, l_w and l_b, such that they satisfy the shared-path-protection constraints with respect to the existing lightpaths?

Theorem 1: DSPLP is NP-complete.

Proof: Please refer to Appendix I. ■

[4]In the wavelength-continuous case, we would associate a conflict set to a wavelength. The conflict set defined here is similar to the conflict vector in [16], the aggregated square matrix in [13], and the "bucket" link metric in [21], but it is more general in the sense that the conflict set can model wavelength-continuous networks, wavelength-convertible networks, and networks of sparse wavelength-conversion capability [9].

[5]The working and backup paths should be link disjoint to protect against link failures and they should also be node disjoint to protect against node failures. For this paper, we shall focus on link disjointedness since link failures (fiber cuts) are the predominant form of failures in telecom networks, while several forms of node failures are combatted through $1 + 1$ redundancy in the nodal hardware components.

III. Compute a Feasible Solution (CAFES)

As the existence version of the problem is NP-complete, we resort to heuristics. In this section, we design a backtracking-based heuristic, called CAFES, to compute an eligible pair of working and backup paths for a lightpath request. In Section IV, we develop a general optimization procedure, called OPT, to iteratively optimize the resource consumption of the working and backup paths for a given solution (i.e., two link-disjoint paths).

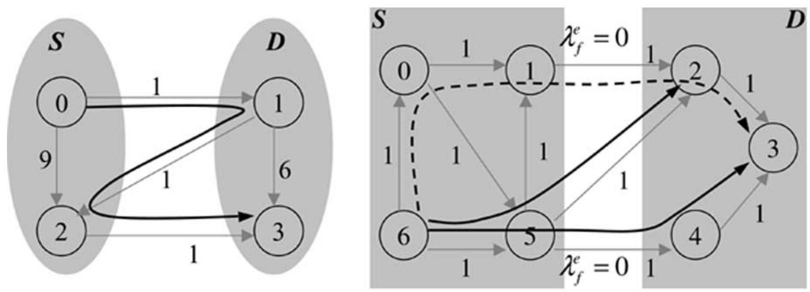
A widely used approach for computing a feasible solution is the so-called two-step approach, which first computes a least-cost path as the working path and then computes as the backup path a link (or node) disjoint path of least additional cost. A limitation of the two-step approach is that it cannot find a solution in a trap topology [6], which is elaborated below, even though a solution exists. An improvement is to compute K working/backup path pairs (typically by applying a K-shortest-path algorithm to compute K candidate working paths and computing a backup path for each candidate working path) and select the pair of minimal cost. As the candidate working paths of the K shortest path pairs may share some common links, enumerating paths with practical values of K, e.g., 2 or 3, may be susceptible to a trap topology as well. Meanwhile, a nontrap topology can later become a trap topology after all the wavelengths on some links are used up. Furthermore, due to backup sharing, trap situations can arise even though the topology is not a trap topology. A possible drawback of enumerating paths is the lack of backtracking, i.e., the information gathered from enumerating the first i paths is not utilized in enumerating the $(i+1)^{th}$ path. We analyze the characteristics of two types of trap situations—trap topology and backup-sharing-caused trap—and propose a backtracking-based solution.

A. Trap Topology

For the example network in Fig. 1(a), the two-step approach cannot find two link-disjoint paths from node 0 to node 3 (even though they exist) because the graph is disconnected after the removal of the working path (which is $\langle 0, 1, 2, 3 \rangle$).

We introduce backtracking based on network flow to overcome the trap situation. Let S be the set of nodes reachable from the source node after removing the links which are not link disjoint to the working path. Let D be the complement of S. (S, D) is referred to as a *cut*. We refer to a link as a *backhaul* link with respect to cut (S, D) if its source node is in D and its destination node is in S. For example, link $\langle 1, 2 \rangle$ in Fig. 1(a) is a backhaul link.

From the viewpoint of network flow, the objective of route computation is to push two units of link-disjoint flow from node s to node d. When a backup path cannot be found, it implies that no flow link-disjoint to the working path can be pushed from set S to set D. If the working path does not use any backhaul link, then clearly there is no link-disjoint flow from node s to node d. However, if the working path does use backhaul links, then it needs to traverse multiple links whose upstream nodes are in set S and downstream nodes are in set D. For example, in Fig. 1(a), the working path traverses two such links $\langle 0, 1 \rangle$ and $\langle 2, 3 \rangle$. The backup path cannot use those links due to the link-disjoint constraint. As a result, no link-disjoint flow can be pushed from set S to set D and no link-disjoint backup path can be found.



(a) (b)

Fig. 1. Trap situations. Solid gray lines represent links; solid black lines denote working paths; dashed black lines denote backup paths; and the number besides a link represents the cost of that link. (a) Trap topology. (b) Backup-sharing-caused trap.

In a two-step approach, working paths can traverse backhaul links because the two-step approach is greedy in the sense that it always selects a least-cost path as the working path. Meanwhile, due to the lack of backtracking in computing the backup path, a two-step approach cannot avoid backhaul links intelligently and, as a result, some lightpath requests may get blocked.

In our proposed approach CAFES, if a backup path is not found, it identifies the set of backhaul links, increases the cost of the backhaul links to some large value such as the sum of the costs of all the links in the network, and restarts the two-step process. This way, the working path will avoid, if possible, these backhaul links, and the backup path will have a chance to reach nodes in D. For example, if we artificially increase the cost of the backhaul link $\langle 1,2\rangle$ to 1000 in Fig. 1(a) and recompute the working path, which turns out to be $\langle 0,1,3\rangle$, we are able to compute a link-disjoint backup path $\langle 0,2,3\rangle$.

B. Backup-Sharing-Caused Trap

Another form of trap situation is termed backup-sharing-cause trap, which is illustrated by using the following example. Consider the network state in Fig. 1(b). One existing lightpath with working path $\langle 6,5,4,3\rangle$ and backup path $\langle 6,0,1,2,3\rangle$ is shown (other existing lightpaths are not shown). Suppose, to protect one more working path traversing link $\langle 6,5\rangle$, links $\langle 5,4\rangle$ and $\langle 1,2\rangle$ both need to allocate one more free wavelength (i.e., $\nu^{\langle 6,5\rangle}_{\langle 5,4\rangle} = \nu^*_{\langle 5,4\rangle}$ and $\nu^{\langle 6,5\rangle}_{\langle 1,2\rangle} = \nu^*_{\langle 1,2\rangle}$) and these two links have no free wavelength. Assume other links have free wavelengths and the cost of each link is unity. When a new lightpath request from node 6 to node 2 comes, a two-step approach may compute path $\langle 6,5,2\rangle$ as the working path. As a result, no backup can be found because no more link-disjoint flow can be pushed from S to D (please note: $\langle 6,5,2\rangle$ and $\langle 6,5,4,3\rangle$ are not link disjoint; $\nu^{\langle 6,5\rangle}_{\langle 5,4\rangle} = \nu^*_{\langle 5,4\rangle}$; $\lambda_f^{\langle 5,4\rangle} = 0$; $\nu^{\langle 6,5\rangle}_{\langle 1,2\rangle} = \nu^*_{\langle 1,2\rangle}$; and $\lambda_f^{\langle 1,2\rangle} = 0$).

Again, we use backtracking to overcome this situation. Define a link $\langle m,n\rangle$ as a conflicting link with respect to cut (S,D) if there exists link $\langle p,q\rangle$ ($p \in S$, $q \in D$) such that $\nu^{\langle m,n\rangle}_{\langle p,q\rangle} = \nu^*_{\langle p,q\rangle}$ and $\lambda_f^{\langle p,q\rangle} = 0$. For example, link $\langle 6,5\rangle$ in Fig. 1(b) is a conflicting link as $\nu^{\langle 6,5\rangle}_{\langle 5,4\rangle} = \nu^*_{\langle 5,4\rangle}$ and $\lambda_f^{\langle 5,4\rangle} = 0$.

If the second minimal-cost path is not found and conflicting links exist, CAFES increases the cost of the conflicting links to some large value and restarts the two-step process. For example, if we artificially increase the cost of the conflicting link $\langle 6,5\rangle$ to 1000 in Fig. 1(b) and recompute the first minimal-cost (working) path, which turns out to be $\langle 6,0,5,2\rangle$, we are able to compute a link-disjoint minimal-cost backup path $\langle 6,5,1,2\rangle$ as it can share the wavelength-link $\langle 1,2\rangle$ with the existing backup $\langle 6,0,1,2,3\rangle$.

If there exist chained trap situations, in which some traps do not appear until some others are processed, we can recursively apply this procedure. We introduce a parameter k to limit the number of recursions. The parameter k can be considered as the maximum number of trap situations we want to process.

A formal specification of our heuristic, CAFES, is in Algorithm 1. In the algorithm, ϵ is a small number, e.g., 10^{-4}. The backup cost function C_1 is used to meet the shared-path-protection constraints C.1–C.4 (the first and last cases in C_1's definition) and to increase backup sharing (the second case). The last case of C_1's definition, $C(e) + \epsilon \cdot (\lambda(e) - \lambda_f^e) \cdot C(e)$, is used for load balancing: when there are two eligible backup paths of the same cost, the less loaded path will be chosen as backup. (Recall that $C(e)$ is the cost of link e.)

Algorithm 1 CAFES

Input: $G = (V,E,C,\lambda)$, $\nu = \{\nu_e | e \in E\}$, s, $d \in V$, k

Output: Two paths l_w and l_b satisfying constraints C.1–C.4, or NULL if no such paths are found.

1) $l'_w \leftarrow \mathrm{NULL}$;
2) compute a minimal-cost path l_w on G from node s to node d; return NULL if l_w is not found or if $l'_w = l_w$;
3) compute a minimal-cost path l_b from node s to node d using cost function:

$$C_1(e) := \begin{cases} +\infty & \text{if } e \in l_w \vee \left(\lambda_f^e = 0 \wedge \left(\exists e' \in l_w, \nu_e^{e'} = \nu_e^*\right)\right) \\ \epsilon \times C(e) & \text{if } \forall e' \in l_w, \nu_e^{e'} < \nu_e^* \\ C(e) + \epsilon \cdot \left(\lambda(e) - \lambda_f^e\right) \cdot C(e) & \text{otherwise} \end{cases}$$

return $\langle l_w, l_b\rangle$ if l_b is found; return NULL if l_b is not found and $k = 0$;
4) compute the set of backhaul links L_b and the set of conflicting links L_c,
5) increase the cost of any link in L_b and L_c to some large value; and
6) $k \leftarrow k - 1$, $l'_w \leftarrow l_w$; go to Step 2.

The computational complexity of CAFES is $O(k \times |E|^2)$. In particular, the complexities of Steps 1–6 are $O(1)$, $O(|V|^2)$, $O(|E|^2)$, $O(|E|^2)$, $O(|E|)$, and $O(1)$, respectively; Steps 2–6 repeat for at most $k+1$ times.

IV. Optimization (OPT)

Given a feasible solution l_w and l_b for a lightpath from node s to node d, we develop a heuristic, called OPT, to minimize the total cost of l_w and l_b, $C_w(l_w)+C_b(l_w,l_b)$. Similar to Algorithm L in [5], OPT iteratively refines l_w and l_b. In one iteration, OPT first recomputes l_w with l_b fixed, and then it recomputes l_b with l_w fixed. The iteration continues as long as there are improvements. How to recompute l_b with l_w fixed is straightforward. We show how to recompute l_w with l_b fixed while jointly optimizing the cost of both. The difficulty here is that *the cost of the backup path l_b changes as the working path l_w changes*.

The basic idea is to consider the changes in backup cost when recomputing the working path l_w. To recompute l_w, we use a standard shortest-path algorithm (such as Dijkstra's algorithm) with a modified *relaxation step* [3] outlined later.

Let H_b be the number of hops l_b has. For any link $e \in E$, associate a backup cost vector $B_e = \{B_e^h | 1 \le h \le H_b\}$. B_e^h denotes the cost of l_b's h^{th} hop if e is used by l_w. B_e^h is defined as follows (e' denotes the h^{th} hop of l_b in the definition shown here)

$$B_e^h := \begin{cases} +\infty & \text{if } \nu_{e'}^e = \nu_{e'}^* \wedge \lambda_f^e = 0 \\ C(e') & \text{if } \nu_{e'}^e = \nu_{e'}^* \wedge \lambda_f^e > 0 \\ \epsilon \cdot C(e') & \text{otherwise.} \end{cases}$$

The first case ($\nu_{e'}^e = \nu_{e'}^* \wedge \lambda_f^e = 0$) indicates that link e' needs to allocate a free wavelength to protect a new working path traversing link e. As there is no free wavelength on link e', link e cannot be used by the working path l_w (l_b is fixed). The second case ($\nu_{e'}^e = \nu_{e'}^* \wedge \lambda_f^e > 0$) implies that link e' needs to allocate a free wavelength (and it is available) to protect a new working path traversing link e. The cost of link e' in this case is the original cost. The last case implies that link e' does not need to allocate any more free wavelength to protect one more working path traversing link e. Thus, link e' is used for "free" if link e is the working path. (Please see Fig. 2.)

We redefine the link-cost function for computing the working path as follows:

$$C_2(e) := \begin{cases} +\infty & \text{if } e \in l_b \vee \lambda_f^e = 0 \\ C(e) & \text{otherwise.} \end{cases}$$

If l_b traverses link e or there is no free wavelength on link e, then the working path cannot utilize link e and its cost is infinite; otherwise, the cost of link e is its original cost.

For any node $v \in V$, associate a cost variable c_w^v and a backup cost vector $B_v = \{B_v^h | 1 \le h \le H_b\}$. c_w^v denotes the cost of the working path from s to v. $\sum_h B_v^h$ indicates the backup cost if the minimum-cost path from s to v is used by the working path l_w. By definition, $C_w(l_w) = c_w^d$ and $C_b(l_w,l_b) = \sum_h B_d^h$.

We then employ a standard shortest-path algorithm to minimize $C_w(l_w) + C_b(l_w,l_b) = c_w^d + \sum_h B_d^h$ with the following relaxation step (please refer to Fig. 2):

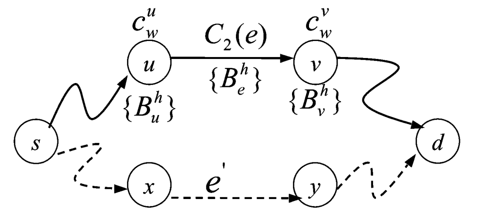


Fig. 2. Illustration of OPT (dashed line is the fixed backup).

```
RELAX(u, v)
  LET B^h = max{B_u^h, B_e^h}, 1 ≤ h ≤ H_b
  IF c_w^u + C_2(e) + Σ_h B^h < c_w^v + Σ_h B_v^h THEN
    c_w^v ← c_w^u + C_2(e)
    B_v^h ← B^h, 1 ≤ h ≤ H_b
    SET NODE u AS NODE v'S PREVIOUS HOP
```

The idea of the above relaxation step is to "relax" along the path which leads to the minimum total cost of the working and the backup paths. To decide whether to use node u as node v's previous hop along the working path (if node v is traversed by the working path), the relaxation step compares the total cost of the working ($c_w^u + C_w(w)$ vs. c_w^u) and the backup ($\sum_h B^h$ vs. $\sum_h B_v^h$). Thus, the changes in backup cost when the working path changes are correctly captured.

A formal specification of OPT is in Algorithm 2. Clearly, if l_w or l_b hits optimum in one iteration, OPT will stop at the next iteration and output the joint optimal l_w and l_b.

Algorithm 2 OPT

Input: $G = (V, E, C_2, \lambda)$, $\nu = \{\nu_e | e \in E\}$, s, $d \in V$, l_w, l_b

Output: Optimized l_w and l_b.

1) compute B_e for $e \in E$ with l_b fixed;
2) $c_w^s \leftarrow 0$, $c_w^v \leftarrow +\infty (v \in V \wedge v \neq s)$;
   $B_s^h \leftarrow 0$, $B_v^h \leftarrow +\infty (1 \le h \le H_b, v \in V \wedge v \neq s)$;
3) compute a minimum-cost path as l_w' using a standard shortest-path algorithm with the new relaxation step shown above;
4) compute a minimum-cost path as l_b' with l_w' fixed using cost function C_1 in Algorithm 1; and
5) if $C_w(l_w') + C_b(l_w', l_b') < C_w(l_w) + C_b(l_w, l_b)$ then $l_w \leftarrow l_w'$, $l_b \leftarrow l_b'$, go to Step 1; otherwise return l_w and l_b.

The computational complexity of OPT is $O(|E|^2)$. In particular, the complexities of Steps 1–5 are $O(|E|^2)$, $O(1)$, $O(|V|^2)$, $O(|E|^2)$, and $O(1)$, respectively; and the number of iterations is typically a small constant for practical-sized networks.

OPT differs from Algorithm L in [5] in the way the working path is computed for a given backup path. Since the cost of a backup path correlates to its working path, both OPT and Algorithm L in [5] consider the changes in backup cost when computing the working path for a given backup path. However, in [5], the changes in backup cost were considered to be *additive* for each hop along the working path. For example, if the first

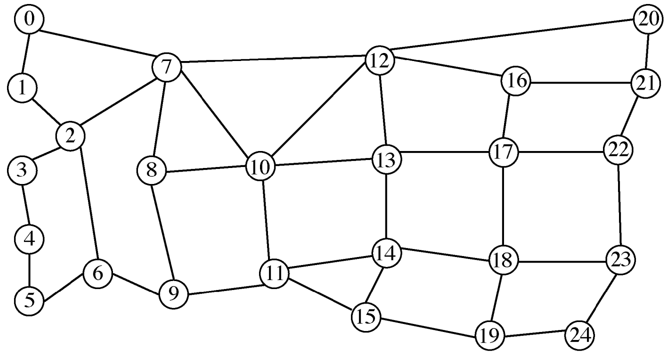


Fig. 3. An example network used in this paper.

hop of a candidate working path incurs c_1 backup cost and the second hop of the candidate working path incurs c_2 backup cost, Algorithm L in [5] considers the changes in backup cost to be c_1+c_2 for the two hops along the candidate working path. However, the changes in backup cost is more meaningful if it is stated as $max\{c_1, c_2\}$, which is the case in OPT. The change from a linear relation, additive, to a nonlinear relation, max, considerably affects how a shortest-path algorithm is modified to compute the working path. In the case of Algorithm L in [5], there is no need to modify a shortest-path algorithm (only to redefine link cost). In the case of OPT, a shortest-path algorithm needs to be modified and a vector of link cost needs to be redefined. Both OPT and Algorithm L in [5] have complexity $O(|E|^2)$, but OPT has a larger constant factor.

V. Illustrative Numerical Results

We now quantitatively evaluate our heuristic algorithms. We simulate a dynamic network environment with the assumptions that the lightpath-arrival process is Poisson and the lightpath-holding time follows a negative exponential distribution. For the illustrative results shown here, in every experiment, 10^6 lightpath requests are simulated; they are uniformly distributed among all node pairs; average lightpath-holding time is normalized to unity; the cost of any link is unity; and our example network topology with 16 wavelengths per fiber is shown in Fig. 3.

We compare our heuristics to an effective two-step approach called Full Information Routing (FIR) [12]. For a lightpath request, FIR first computes a least-cost path as the working path and then computes as backup path a least-cost path using cost function $C_1(e)$ (without the load balancing) based on the working path. The feasible solution to OPT is the solution of CAFES (with $k = 1$ since we found that the performance improvement is marginal if we increase k to any larger value).[6]

A. Blocking Probability

Fig. 4 compares the blocking probability of our heuristics to FIR. We observe that our heuristic has lower blocking probability. This is because of the backtracking in CAFES and the joint optimization in OPT. The reason that the difference between the two curves decreases as the network offered load increases is as follows. When the network offered load is modest or low, a lightpath is unlikely to be blocked because of resource limitation. Blocking happens mainly because a heuristic cannot find two eligible link-disjoint paths. Since FIR fails in trap situations while CAFES does not, FIR has a much higher blocking probability. When the network offered load is high, a lightpath is more likely to be blocked because there are no available wavelengths on some links. While some lightpath requests still get blocked due to trap situations in FIR, other lightpath requests (which might be blocked in CAFES) can utilize the resources these lightpath requests are supposed to use (if they are accepted). Although CAFES can find a solution for a lightpath request with much higher probability than FIR, accepting a lightpath with a "detoured" route (as in the case when FIR cannot find a route while CAFES can) might interfere or even block future lightpath arrivals. Thus, the difference between the two curves decreases (but they will not cross each other because of the joint optimization in OPT).

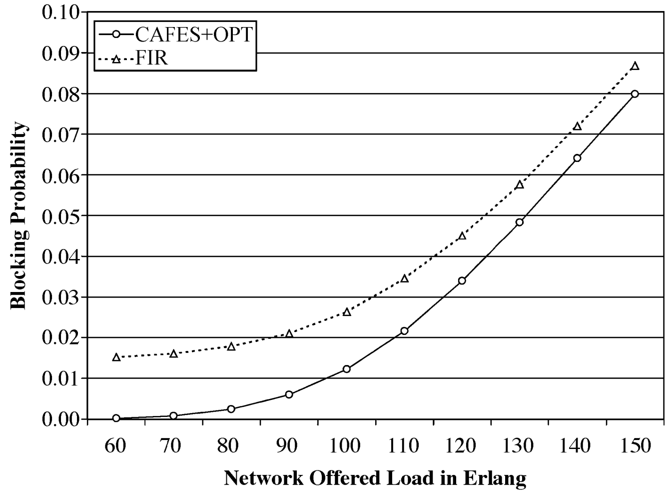


Fig. 4. Blocking probability.

B. Resource Overbuild

One figure of merit for comparing resource efficiency is *resource overbuild*, defined as the amount of wavelength links consumed by backup paths over the amount of wavelength links utilized by working paths [12]. Resource overbuild indicates the amount of extra resources needed for providing protection as the percentage of the amount of resources required without protection. Typically, it is desirable to have lower resource overbuild because lower resource overbuild implies better backup sharing. Fig. 5 shows that our heuristics have lower resource overbuild over FIR. The optimization procedure in OPT contributes to the reduction in resource overbuild.

C. Average Hop Distance

The advantages of our heuristics come for a slight increase in the average backup-path hop distance. Fig. 6 shows that both our heuristics and FIR have similar average working-path hop distance while our heuristics have slight longer average backup-path hop distance. The load balancing in CAFES and

[6]While we use CAFES to generate feasible solutions as input for OPT here, we can use any shared-path-protected routing heuristic, e.g., FIR. In fact, we used FIR to generate feasible solutions as input to OPT, and we observed modest reduction in blocking probability. Thus, OPT is complementary to existing shared-path-protected routing approaches.

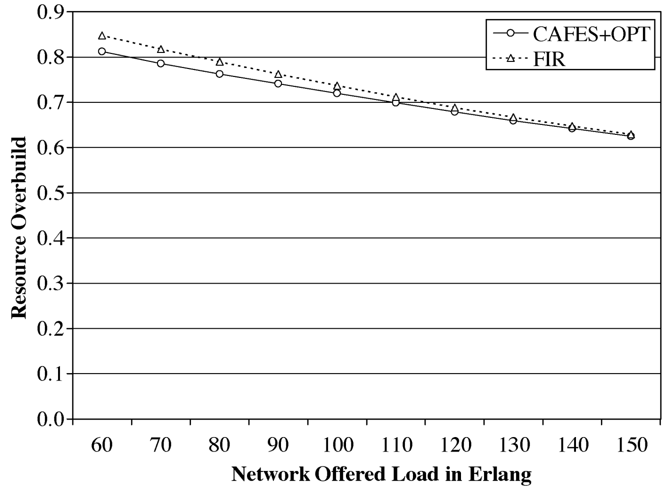


Fig. 5. Resource overbuild.

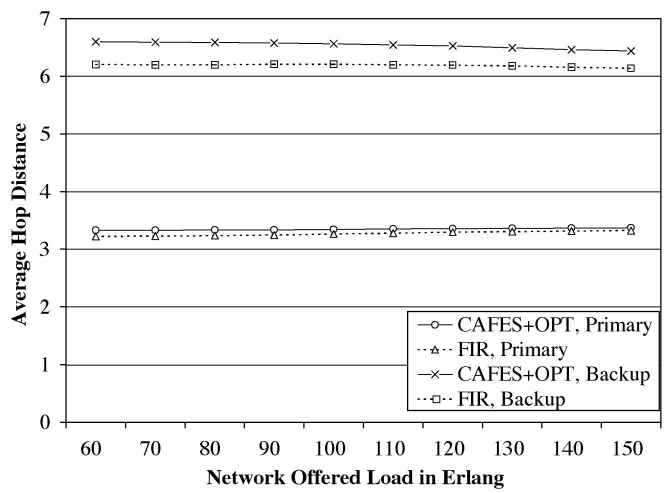


Fig. 6. Average hop distance.

the joint optimization in OPT (which tends to maximize backup shareability) lead to the increase in backup hop distance. Figs. 4, 5, and 6 imply that our heuristics lead to better backup sharing than FIR does because: 1) the average backup-path hop distance of our heuristics is longer than that of FIR; 2) our heuristics still have lower resource overbuild; and 3) and our heuristics have lower blocking probability.

VI. Conclusion

This paper is devoted to dynamic shared-path-protected lightpath provisioning in optical WDM mesh networks. We proved that the problem of finding an eligible pair of working and backup paths under shared-path-protection constraints in a WDM mesh network for a new lightpath request with respect to existing lightpaths is NP-complete. We presented a heuristic, called CAFES, to compute a feasible solution and another heuristic, called OPT, to optimize the resource consumption for a given solution. Simulation results demonstrated that our heuristic algorithms are efficient and effective. While our focus here is on dynamic lightpath provisioning in WDM context, the core ideas of how to compute a feasible solution and how to jointly optimize a given solution can be applied to: 1) restorable bandwidth-guaranteed-connection provisioning in MPLS networks; and 2) and static shared-path-protected lightpath provisioning in which the traffic matrix is known *a priori*.

Appendix I
NP-Completeness of Dynamic Shared-Path-Protected Lightpath-Provisioning (DSPLP) Problem

Basic Idea on Our Proof of Theorem 1: We reduce 3SAT, which is known to be NP-complete [8], to the Dynamic Shared-Path-Protected Lightpath-Provisioning (DSPLP) problem. The 3SAT problem is formally stated as follows.

3SAT Instance: Collection $F = \{D_1, D_2, \ldots, D_M\}$ of clauses on a finite set Q of variables such that $|D_i| = 3$ for $1 \leq i \leq M$, where clause D_i is the Boolean "or" of three literals (a literal is either a variable or the Boolean "not" of a variable) and is satisfied by a truth assignment if and only if at least one of the three literals is true. $|F| = M$, where M is the number of clauses; $|Q| = N$, where N is the number of variables.

3SAT Question: Is there a truth assignment for Q that satisfies all the clauses in F?

The basic idea of our proof is, for an arbitrary instance of 3SAT, to construct a graph such that there is a path l_w corresponding to the truth assignment of all the clauses and a path l_b corresponding to the false assignment of all the variables. The two paths satisfy the shared-path-protection constraints (Constraints C.1–C.4) if and only if the given instance of 3SAT is satisfied. The formal proof follows.

Proof of Theorem 1: DSPLP $\in$ NP since a nondeterministic algorithm can guess two paths l_w and l_b and check in polynomial time if these two paths satisfy Constraints C.1–C.4 with respect to existing lightpaths in $\mathcal{L}$.

Given an arbitrary instance of 3SAT $F = \{D_1, D_2, \ldots, D_M\}$ and $Q = \{v_1, v_2, \ldots, v_N\}$, we construct in polynomial time an instance of DSPLP $G = (V, E, C, \lambda)$, $\mathcal{L}$, s, and d where $C(e) = 1$ and $\lambda(e) = 1$ for all $e \in E$. In the following construction, we first define the set of nodes V, then define the set of links E, and finally define the set of existing lightpaths $\mathcal{L}$.

The set of nodes V consists of a source node s and a destination node d. The other nodes in V can be divided into three groups. The first group is related to the variables in Q and consists of the nodes a_k, b_j^i, $\bar{b}_j^i$, c_j^i, and $\bar{c}_j^i$, where $0 \leq k \leq N$, $1 \leq i \leq M$, $1 \leq j \leq N$. The second group of nodes is related to the clauses in F and consists of the nodes p_i and q_i for $1 \leq i \leq M$. The third group of nodes is the source nodes of the existing lightpaths and consists of z_i for $0 \leq i \leq N$.

The set of links (unidirectional fibers) E can also be divided into three groups. In the reduction, the first group of links will be used for assigning values to the variables in Q; the second group of links will be used for evaluating the Boolean value of the clauses; the third group of links will be used for existing lightpaths. The three groups of links are as follows.

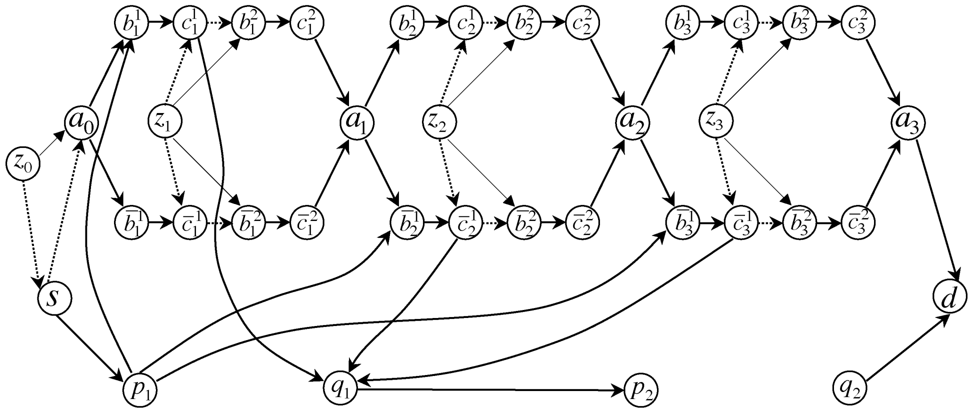


Fig. 7. An illustrative construction for a 3SAT instance $F = \{D_1, D_2\}, D_1 = v_1 \vee \bar{v}_2 \vee \bar{v}_3, D_2 = \bar{v}_1 \vee v_2 \vee v_3$, and $Q = \{v_1, v_2, v_3\}$ (the part corresponding to clause D_2 is not shown). A dotted line corresponds to a link whose only wavelength has been used by an existing backup path. A thin solid line corresponds to a link whose only wavelength has been used by an existing working path. A thick solid line corresponds to a link whose only wavelength is available.

— Group 1:
  a) A link from node b_j^i to node c_j^i, $1 \le i \le M$, and $1 \le j \le N$.
  b) A link from node $\bar{b}_j^i$ to node $\bar{c}_j^i$, $1 \le i \le M$, and $1 \le j \le N$.
  c) A link from node c_j^i to node b_j^{i+1}, $1 \le i \le M-1$, and $1 \le j \le N$.
  d) A link from node $\bar{c}_j^i$ to node $\bar{b}_j^{i+1}$, $1 \le i \le M-1$ and $1 \le j \le N$.
  e) A link from node a_j to node b_{j+1}^1, and a link from node a_j to node $\bar{b}_{j+1}^1$, $0 \le j \le N-1$.
  f) A link from node c_j^M to node a_j, and a link from node $\bar{c}_j^M$ to node a_j, $1 \le j \le N$.

—) Group 2:
  a) A link from node p_i to b_j^i, and a link from node c_j^i to node q_i, if and only if the variable v_j is in clause D_i.
  b) A link from node p_i to $\bar{b}_j^i$, and a link from node $\bar{c}_j^i$ to node q_i, if and only if $\bar{v}_j$ is in clause D_i.
  c) A link from node q_i to node p_{i+1}, $1 \le i \le M-1$.

—) Group 3:
  a) A link from z_0 to s and a link from z_0 to a_0.
  b) A link from z_j to c_j^i and a link from z_j to $\bar{c}_j^i$, where $1 \le i < M$ and $1 \le j \le N$.
  c) A link from z_j to b_j^i and a link from z_j to $\bar{b}_j^i$, where $1 < i \le M$ and $1 \le j \le N$.

In addition to these links, we also have a link from node s to node a_0, a link from node a_N to node d, a link from node s to node p_1, and a link from node q_M to node d. It is easy to see that the construction can be done in polynomial time.

The set of $2 \times (M-1) \times N + 1$ existing lightpaths $\mathcal{L}$ is defined as follows:

1) Lightpath 1: working path $\langle z_0, a_0 \rangle$, backup path $\langle z_0, s, a_0 \rangle$.
2) Lightpaths 2 to $(M-1) \times N+1$: working path $\langle z_j, b_j^{i+1} \rangle$, backup path $\langle z_j, c_j^i, b_j^{i+1} \rangle$, where $1 \le i < M$ and $1 \le j \le N$.
3) Lightpaths $(M-1) \times N + 2$ to $2 \times (M-1) \times N + 1$: working path $\langle z_j, \bar{b}_j^{i+1} \rangle$, backup path $\langle z_j, \bar{c}_j^i, \bar{b}_j^{i+1} \rangle$, where $1 \le i < M$ and $1 \le j \le N$.

An illustrative construction (the construction for the second clause is similar to the first one and thus it is not shown) for a 3SAT instance $F = \{D_1, D_2\}$, $D_1 = v_1 \vee \bar{v}_2 \vee \bar{v}_3$, $D_2 = \bar{v}_1 \vee v_2 \vee v_3$, and $Q = \{v_1, v_2, v_3\}$ is shown in Fig. 7.

We now show that, if F is satisfiable, then from node s to node d in graph G there exist two paths satisfying the shared-path-protection constraints C.1–C.4 with respect to the existing lightpaths $\mathcal{L}$. Let $v_1 = x_1, v_2 = x_2, \ldots, v_N = x_N$ be an assignment that satisfies F, where $x_j \in \{0, 1\}$ for $1 \le j \le N$. The two paths can be routed as follows. The backup path l_b is routed via the nodes b_j^i and c_j^i ($1 \le j \le N$ and $1 \le i \le M$) if and only if $x_j = 0$; otherwise, l_b is routed via nodes $\bar{b}_j^i$ and $\bar{c}_j^i$ (l_b will also need to traverse all the a_k nodes for $0 \le k \le N$). The working path l_w is routed via p_i, q_i, and other nodes defined as follows. By the construction, link $\langle p_i, q_i \rangle$ corresponds to clause D_i, $|D_i| = 3$. Without loss of generality, let $D_i = v_f \vee \bar{v}_g \vee v_h$, where f, g, and h are distinct integers between 1 and N. Then, there will be three paths from node p_i to node q_i that go through the nodes b_f^i, $\bar{b}_g^i$, and b_h^i, respectively. Since $v_1 = x_1$, $v_2 = x_2, \ldots, v_N = x_N$ is an assignment that satisfies F, either $x_f = 1$, or $x_g = 0$, or $x_h = 1$. If $x_f = 1$, then the working path traverses nodes b_f^i and c_f^i; if $x_g = 0$, then the working path traverses nodes $\bar{b}_g^i$ and $\bar{c}_g^i$; if $x_h = 1$, then the working path traverses nodes b_h^i and c_h^i; if more than one condition is true, then randomly pick one. l_w and l_b so selected are link disjoint (Condition C.1) because of the following facts:

1) l_w traverses links in Group 1 and Group 2.
2) l_b only traverses links in Group 1.
3) The Group-1 links l_w traverses correspond to the literals of value 1. We will use the term "1-value link" ("0-value link") to denote a link corresponding to a literal of value 1 (0).
4) The Group-1 links l_b traverses correspond to the literals of value 0.
5) A Group-1 1-value link is disjoint to a Group-1 0-value link by the construction.
6) A Group-1 link is disjoint to a Group-2 link by the construction.

The Condition C.2 holds because: l_w uses links in Group 1 and Group 2 while any working path in $\mathcal{L}$ only utilizes links in Group 3; a link in Group 3 is disjoint to a link in either Group 1 or

Group 2. The Condition C.3 holds because: any backup path in $\mathcal{L}$ utilizes 0-value links in Group 1 and links in Group 3; l_w uses 1-value links in Group 1 and links in Group 2; a link in Group 2 is disjoint to a link in either Group 1 or Group 3; a 0-value link in Group 1 is disjoint to a 1-value link in Group 1. The Condition C.4 holds because l_w does not traverse the same SRLG as any working path in $\mathcal{L}$ (thus, l_b can share any wavelength-link with any backup path in $\mathcal{L}$) due to the facts that: l_w only utilizes links in Group 1 and Group 2; any working path in $\mathcal{L}$ only utilizes links in Group 3; a link in Group 3 is disjoint to a link in either Group 1 or Group 2. Thus, we have shown that, if F is satisfiable, then we can find for the lightpath request from s to d two lightpaths satisfying shared-path-protection constraints with respect to the existing lightpaths.

Suppose from s to d in the constructed graph G there exist two paths (working path l_w and backup path l_b) satisfying the shared-path-protection constraints (Constraints C.1–C.4) with respect to the existing lightpaths $\mathcal{L}$. We show that F is satisfiable using the following facts:

1) Since there are only two links emanating from the source node s, one of them will be used by the working path l_w while the other will be used by the backup path l_b due to Constraint C.1 that l_w and l_b are link-disjoint.
2) The working path l_w is routed via the p_i and q_i nodes (and some Group-1 nodes which will be specified later) because of Constraint C.2 and the construction that the only wavelength on link $\langle s, a_0 \rangle$ is used by an existing backup path.
3) The backup path l_b traverses links only in Group 1 and all the a_j nodes. If the backup path l_b uses any link in Group 2, then l_b will have to traverse some link $\langle q_i, p_{i+1} \rangle$ for some $1 \leq i < M$, which is used by the working path. Thus, the two lightpaths will violate Constraint C.1. Clearly l_b cannot use any link in Group 3. Furthermore, if, for some value of j, the backup path traverses node b_j^i ($\overline{b}_j^i$), then it must traverse all the $b_j^i(\overline{b}_j^i)$ nodes and the $c_j^j(\overline{c}_j^i)$ nodes, where $1 \leq i \leq M$.
4) We represent the route for the backup path l_b by a N-bit binary number $\beta_1\beta_2\ldots\beta_N$. The backup path is routed as follows: if bit β_j is 1, then the backup path is routed over the nodes $\overline{b}_j^i$; otherwise, it is routed over the nodes b_j^i. Note that there is a one-to-one mapping between an N-bit binary number and a backup route in the network. Let $\beta_1\beta_2\ldots\beta_N$ be the N-bit number that corresponds to the backup path. Then, there is a path from node p_i to node q_i that does not use the links in the backup path, if and only if clause D_i is true under the following assignment: $v_j = \beta_j$, where $1 \leq j \leq N$. The reason is as follows. Without loss of generality, let $D_i = v_f \vee \overline{v}_g \vee v_h$, where f, g, and h are distinct integers between 1 and N. Note that there are three paths from node p_i to node q_i that traverse the nodes b_f^i, $\overline{b}_g^i$, and b_h^i, respectively. The backup path will traverse the nodes b_f^i, $\overline{b}_g^i$, and b_h^i, if and only if $\beta_f = 0$, $\beta_g = 1$, and $\beta_h = 0$, respectively. Thus, there exists a path from node p_i to q_i if and only if β_f is 1, or β_g is 0, or β_h is 1, which is exactly the condition under which D_i will be true.

Since the working path and the backup path satisfy the shared-path-protection constraints, D_i is true for all $1 \leq i \leq M$ due to the above facts. Thus, F is satisfiable under the following assignment: $v_j = \beta_j$, $1 \leq j \leq N$, where $\beta_1\beta_2\ldots\beta_N$ is the N-bit number corresponding to the backup path l_b.

This concludes our proof that DSPLP is NP-complete.

References

[1] E. Bouillet, J.-F. Labourdette, G. Ellinas, R. Ramamurthy, and S. Chaudhuri, "Stochastic approaches to compute shared mesh restored lightpaths in optical network architectures," in *Proc. IEEE INFOCOM*, June 2002, pp. 801–807.

[2] E. Bouillet, J.-F. Labourdette, R. Ramamurthy, and S. Chaudhuri, "Enhanced algorithm cost model to control tradeoffs in provisioning shared mesh restored lightpaths," in *Proc. OFC*, Mar. 2002, p. ThW2.

[3] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*. New York: McGraw-Hill, 1990.

[4] O. Crochat, J.-Y. Le Boudec, and O. Gerstel, "Protection interoperability for WDM optical networks," *IEEE/ACM Trans. Networking*, vol. 8, pp. 384–395, June 2000.

[5] B. T. Doshi, S. Dravida, P. Harshavardhana, O. Hauser, and Y. Wang, "Optical network design and restoration," *Bell Labs Tech. J.*, vol. 4, pp. 58–84, Jan.–Mar. 1999.

[6] D. Dunn, W. Grover, and M. MacGregor, "Comparison of k-shortest paths and maximum flow routing for network facility restoration," *IEEE J. Select. Areas Commun.*, vol. 12, pp. 88–99, Jan. 1994.

[7] D. Elie-Dit-Cosaque, M. Ali, and L. Tancevski, "Informed dynamic shared path protection," in *Proc. OFC*, Mar. 2002, p. ThO4.

[8] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York: W. H. Freeman, 1979.

[9] J. Iness and B. Mukherjee, "Sparse wavelength conversion in wavelength-routed WDM networks," *J. Photonic Network Commun.*, vol. 1, pp. 183–205, Nov. 1999.

[10] M. Kodialam and T. V. Lakshman, "Dynamic routing of bandwidth guaranteed tunnels with restoration," in *Proc. IEEE INFOCOM*, vol. 2, Mar. 2000, pp. 902–911.

[11] ——, "Dynamic routing of locally restorable bandwidth guaranteed tunnels using aggregated link usage information," in *Proc. IEEE INFOCOM*, vol. 1, Apr. 2001, pp. 376–385.

[12] G. Li, D. Wang, C. Kalmanek, and R. Doverspike, "Efficient distributed path selection for shared restoration connections," in *Proc. IEEE INFOCOM*, June 2002, pp. 140–149.

[13] Y. Liu, D. Tipper, and P. Siripongwutikorn, "Approximating optimal spare capacity allocation by successive survivable routing," in *Proc. IEEE INFOCOM*, vol. 2, Apr. 2001, pp. 699–708.

[14] S. S. Lumetta, M. Medard, and Y.-C. Tseng, "Capacity versus robustness: a tradeoff for link restoration in mesh networks," *J. Lightwave Technol.*, vol. 18, pp. 1765–1775, Dec. 2000.

[15] E. Modiano and A. Narula-Tam, "Survivable lightpath routing: A new approach to the design of WDM-based networks," *IEEE J. Selected Areas in Communications*, vol. 20, pp. 800–809, May 2002.

[16] G. Mohan, C. S. R. Murthy, and A. K. Somani, "Efficient algorithms for routing dependable connections in WDM optical networks," *IEEE/ACM Trans. Networking*, vol. 9, pp. 553–566, Oct. 2001.

[17] C. Qiao and D. Xu, "Distributed Partial Information Management (DPIM) schemes for survivable networks—Part I," in *Proc. IEEE INFOCOM*, June 2002, pp. 302–311.

[18] R. Ramamurthy, Z. Bogdanowicz, S. Samieian, D. Saha, B. Rajagopalan, S. Sengupta, S. Chaudhuri, and K. Bala, "Capacity performance of dynamic provisioning in optical networks," *J. Lightwave Technol.*, vol. 19, pp. 40–48, Jan. 2001.

[19] S. Ramamurthy, L. Sahasrabuddhe, and B. Mukherjee, "Survivable WDM mesh networks," *J. Lightwave Technol.*, vol. 21, pp. 870–883, Apr. 2003.

[20] C. Su and X. Su, "Protection path routing on WDM networks," in *Proc. OFC*, vol. 2, Mar. 2001, pp. TuO2-T1–TuO2-T3.

[21] X. Su and C. Su, "An online distributed protection algorithm in WDM networks," in *Proc. IEEE ICC*, vol. 5, June 2001, pp. 1571–1575.

[22] J. W. Suurballe and R. E. Tarjan, "A quick method for finding shortest pairs of disjoint paths," *Networks*, no. 14, pp. 325–336, 1984.

[23] B. Van Caenegem, W. Van Parys, F. De Turck, and P. Demeester, "Dimensioning of survivable WDM networks," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 1146–1157, Sept. 1998.

[24] C. Xin, Y. Ye, S. Dixit, and C. Qiao, "A joint working and protection path selection approach in WDM optical networks," in *Proc. IEEE Globecom*, vol. 4, 2001, pp. 2165–2168.

[25] Y. Xiong, D. Xu, and C. Qiao, "Achieving fast and bandwidth-efficient shared-path protection," *IEEE J. Lightwave Technology*, vol. 21, pp. 365–371, Feb. 2003.

Canhui (Sam) Ou (S'02) is currently working toward the Ph.D. degree with the Computer Science Department, University of California, Davis.

His research interests include design and analysis of wavelength-routed wavelength-division-multiplexed networks with focus on survivability and traffic grooming.

Jing Zhang (S'02) is working toward the Ph.D. degree with the Computer Science Department, University of California, Davis.

Her research interests include design and analysis of wavelength-routed wavelength-division-multiplexed netwoks with focus on survivability and availability.

Hui Zang (S'97–M'02) received the B.S. degree in computer science from Tsinghua Unversity, Beijing, China, in 1997 and the M.S. and Ph.D. degrees in computer science from the University of California, Davis, in 1998 and 2001, respectively.

She joined Sprint Advanced Technology Laboratories, Burlingame, CA, in 2000, where she is a Principal Applied Research and Development Engineer. She is the author of the book *WDM Mesh Networks—Management and Survivability* (Norwell, MA: Kluwer, 2002). Her research interests include wavelength-division-multiplexed (WDM) optical networks, Internet protocol (IP) over WDM networks, multiservice platforms, and Quality of Service.

Dr. Zang was one of the Guest Editors of the IEEE NETWORK Special Issue on Traffic Engineering in Optical Networks.

Laxman H. Sahasrabuddhe (S'94–M'00) received the B.Tech. degree from the Indian Institute of Technology, Kanpur, in 1992, the M.Tech. degree from the Indian Institute of Technology, Madras, in 1994, and the Ph.D. degree from the University of California, Davis, in 1999.

From 1999 to 2000, he was an Embedded Software Engineer at Amber Networks, which was acquired by Nokia in July 2001. Currently, he is a Principal Member of Technical Staff at SBC Services, Inc., San Ramon, CA.

Dr. Sahasrabuddhe is the receipient of the "Best Doctoral Dissertation Award," from the College of Engineering, University of California, for his research on wavelength-division-multiplexing (WDM) optical networks.

Biswanath Mukherjee (S'82–M'87) received the B.Tech. degree (with honors) from the Indian Institute of Technology, Kharagpur, in 1980 and the Ph.D. degree from University of Washington, Seattle, in June 1987.

He held a GTE Teaching Fellowship and a General Electric Foundation Fellowship with the University of Washington. In July 1987, he joined the University of California, Davis, where he has been Professor of Computer Science since July 1995 and served as Chairman of Computer Science from September 1997 to June 2000. He serves or has served on the editorial boards of the *ACM/Baltzer Wireless Information Networks* (WINET), *Journal of High-Speed Networks*, *Photonic Network Communications*, and *Optical Network Magazine*. He is author of the textbook *Optical Communication Networks* (New York: McGraw-Hill, 1997), a book that received the Association of American Publishers, Inc.'s 1997 Honorable Mention in Computer Science. He is a Member of the Board of Directors of IPLocks, Inc., a Silicon Valley startup company. He has consulted for and served on the Technical Advisory Board of a number of startup companies in optical networking. His research interests include lightwave networks, network security, and wireless networks.

Dr. Mukherjee is co-winner of paper awards presented at the 1991 and the 1994 National Computer Security conferences. He serves or has served on the editorial boards of the IEEE/ACM TRANSACTIONS ON NETWORKING and IEEE *Network*. He also served as Editor-at-Large for optical networking and communications for the IEEE Communications Society and as the Technical Program Chair of the IEEE INFOCOM '96 conference.