On Disjoint Path Pairs with Wavelength Continuity Constraint in WDM Networks

Reid Andersen  Fan Chung
Department of Computer Science and Engineering
University of California, San Diego, CA 90005
Email: \{reid, fan\}@ucsd.edu

Arunabha Sen  Guoliang Xue
Department of Computer Science and Engineering
Arizona State University, Tempe, AZ 85287
Email: \{asen, xue\}@asu.edu

Abstract—In a WDM optical network, each fiber link can carry a certain number of wavelengths \( \Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_W\} \). To tolerate a single link (node, respectively) failure in the network, the path protection scheme of fault management establishes an active path and a link-disjoint (node-disjoint, respectively) backup path, so that in the event of a link failure (node failure, respectively) on the active path, data can be quickly re-routed through the backup path. We consider a dynamic scenario, where requests to establish active-backup paths between a specified source-destination node pair arrive sequentially. If a link-disjoint (node-disjoint, respectively) active-backup path pair can be found at the time of the request, the paths are established; otherwise, the request is blocked. In this scenario, at the time a request arrives, not every fiber link will have all \( W \) wavelengths available for new call establishment, as some of the wavelengths may already have been allocated to earlier requests and communication through these paths may still be in progress. We also assume that the network nodes do not have any wavelength converters. This paper studies the existence of a pair of link-disjoint (node-disjoint, respectively) active-backup paths satisfying the wavelength continuity constraint between a specified source-destination node pair. First we prove that both the link-disjoint and node-disjoint versions of the problem are NP-Complete. Then we concentrate on the link-disjoint version and present an approximate algorithm and an exact algorithm for the problem. Finally, through our experimental evaluations, we demonstrate that our approximate algorithm produces near-optimal solutions in almost all of the instances of the problem in a fraction of the time needed to find the optimal algorithm.

I. INTRODUCTION

Survivability of high bandwidth optical networks has emerged as an important area of research in recent years due to its tremendous importance as a national and international infrastructure for moving large volumes of data. Failure of any part of this infrastructure, either due to natural causes or malicious attacks is bound to have a significantly large adverse impact on the economy. In the last few years researchers have been examining survivability issues in WDM networks [1], [2], [5], [17], [7], [14], [18], [19], [21], [22], [26], [30], [31], [32]. Two techniques, protection at the WDM layer and restoration at the IP layer have emerged as the two main contenders for fault management in optical networks [20], [25]. Protection refers to pre-provisioned failure recovery (usually hardware based) whereas restoration refers to more dynamic recovery (usually software based) [10], [9]. Between the two schemes, protection is typically faster and normally offers single link failure and single node failure protection. The protection schemes can be divided into link protection, path protection [22], [23], and partial path protection schemes [11], [29]. Path protection can be further subdivided into shared path protection and dedicated path protection [22], [23]. In the shared path protection scheme, spare capacity is shared among various backup paths whereas in dedicated path protection, spare capacity is reserved for each individual source-to-destination path and no sharing of this capacity is allowed [12].

Fig. 1. Active and backup paths in a WDM network are link-disjoint.
In optical networks, cuts in fibers are considered to be one of the most common failures, while failures of routers are also possible. In the dedicated path protection scheme, an alternate path is maintained in a stand-by mode for every source-destination path used for data transmission. These paths are referred to as the secondary or backup path and the primary or active path, respectively. Fig. 1 shows a WDM network (with 8 nodes and 10 links) and 2 existing connections. The active and the backup paths between the nodes $a$ and $d$ are $a-b-d$ on wavelength $\lambda_1$ and $a-f-d$ on wavelength $\lambda_2$, respectively. The active and the backup paths between the nodes $f$ and $h$ are $f-g-h$ on wavelength $\lambda_2$ and $f-d-h$ on wavelength $\lambda_1$, respectively. Clearly, in order to tolerate any single link (node, respectively) failure, the backup path should not be sharing any fiber link (node, respectively) with its corresponding active path. Thus the backup path should be link-disjoint (node-disjoint, respectively) with the active path.

In this paper we study the following problem: Consider a WDM network where each fiber can carry $W$ wavelengths $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_W\}$. A lightpath [4], [15] is established between a source-destination node pair when a request for such a path arrives and appropriate network resources are available. For reasons of survivability, following the dedicated path protection strategy, we try to establish a primary (active) as well as a secondary (backup) path. If sufficient network resources are available at the time the request arrives, the active-backup path pair is established, otherwise the call request is blocked. We assume that the nodes do not have wavelength converters and as a consequence, the active path must maintain the same wavelength throughout the entire path. The same is true for the backup path, although it may be using a different wavelength. In the WDM research community, this is known as the wavelength continuity constraint. We point out that computing a pair of active-backup paths with shared protection is more complicated than the case with dedicated protection. Therefore the NP-completeness of the problem with dedicated protection gives a strong indication of the hardness of the problem with shared protection. Consider a situation where a request to establish an active-backup path pair arrives at time $T$. At time $T$, several active-backup path pairs may already be in existence. Accordingly, not every fiber link will have all $W$ wavelengths available for the establishment of the new paths.

The use of wavelength converters is considered expensive in current WDM networks. Also, as shown in the appendix, the disjoint paths problems can be solved in polynomial time in WDM networks with wavelength converters, while the disjoint paths problems are NP-complete in WDM networks without wavelength converters.

It is possible that link 1 may have only wavelengths $\{\lambda_1, \lambda_3\}$ available, link 2 may have only wavelength $\lambda_2$ available, and so on.

We refer to the problem under study in this paper as the Disjoint Path Problem under Wavelength Continuity Constraint (DPPWCC). Informally, it can be stated as follows: Given a set $\Lambda(e) \subseteq \Lambda$ of wavelengths available on each link $e$, is it possible to establish an active and a backup path between a specified source-destination node pair satisfying the wavelength continuity constraint?

As discussed earlier, for link survivability (node survivability, respectively), the active and the backup paths should be link-disjoint (node-disjoint, respectively). Therefore there is a link version (LDPPWCC) and a node version (NDPPWCC) of the DPPWCC problem. The following approach may be considered for solving the DP- PWCC problem. First, consider a network comprising of only those links where wavelength $\lambda_1$ is available for call establishment. If this network has two link-disjoint (node-disjoint, respectively) paths between the specified source-destination node pair, then we have a solution for the DPPWCC problem $^2$. If this network does not have two link-disjoint (node-disjoint, respectively) paths between the specified source-destination node pair, then this process can be repeated for other wavelengths $\lambda_2$ and $\lambda_3$ etc. If one of them finds the disjoint paths, we have a solution for the DPPWCC problem. It may be noted that if any one of the attempts succeeds, both the active and the backup paths will be established using the same wavelength. However, it may be noted that it is not necessary that the active and the backup paths use the same wavelength. It may be possible to establish the active path using wavelength $\lambda_2$ and the backup path using wavelength $\lambda_1$, as in the case of the connection between nodes $f$ and $h$ in Fig. 1. This situation is certainly more complex than the one where both the active path and the backup path use the same wavelength. As will be seen in this paper, under this situation both the LDPPWCC problem and the NDPPWCC problem are intractable.

The objective of this paper is to specifically tackle the DPPWCC problems. We show that the disjoint path computation problem with wavelength continuity constraint is NP-Complete, a commonly held belief in the WDM networking research community. This formal proof validates the study of heuristics and integer linear programming (ILP) formulations. We also design an enhanced version of the commonly used active path first heuristic and present simulation results showing that our new heuristic finds optimal solutions in 99.8% of the cases tested.

$^1$The existence of a pair of disjoint paths on the same wavelength can be solved in polynomial time using Suurballe’s algorithm [27], [28].
while using only a fraction of the time required by the integer linear programming based algorithm. The rest of the paper is organized as follows. Section II provides some background in related areas. Section III states the problems in a formal setting. Section IV presents the complexity result for the problems. Sections V and VI provide an approximate and an exact solution for the link version of the LDPPWCC problem. Section VII compares the results between the exact and the approximate solutions and Section VIII concludes the paper.

II. RELATED WORK

Although several researchers in the last few years have published a significant number of papers on survivability issues in WDM optical networks [1], [2], [5], [17], [7], [14], [16], [18], [21], [22], [26], to the best of our knowledge, the topic of this paper, the complexity of the disjoint paths problem with wavelength continuity constraint, remains open. Although many researchers in the area believe that the problem is NP-Complete [6], [3], there is no formal proof available in the literature. A major contribution of this paper is to provide for the first time a formal proof that the problem indeed is NP-Complete [8].

Recently, Hu [12] published NP-Completeness results related to diverse routing in optical mesh networks [3]. Although at a first glance it may appear that the problems discussed in [12] are the same as the problems discussed in this paper, they are in fact significantly different. In this paper we consider the case where each link can carry only a certain subset of the wavelengths, and ask whether a disjoint pair of active-backup lightpaths satisfying the wavelength continuity constraint can be established between a specified pair of nodes. In the problem studied in [12], the logical topology of the network is given as part of the input. This implies that lightpaths have already been established between the appropriate pairs of nodes. The objective of the problem studied in [12] is not to try establish a new active-backup lightpath pair, but to use the already established lightpaths to find link-disjoint active-backup paths between the specified source-destination node pair, subject to the constraint that failure of a single physical link would not disrupt both the active and the backup paths. Even in the fiber disjoint path problem where only link failures are considered, the objective is not to establish a new pair of disjoint lightpaths, but to find a pair of disjoint paths using the existing lightpaths. Moreover, the wavelength continuity constraint, which plays a key role in the problem under consideration in this paper, has no role in the problem considered in [12].

In order to compute a pair of link-disjoint active-backup lightpaths, the current literature uses what is known as the shortest active path first (APF) heuristic [6], [13], which first computes a shortest lightpath as a candidate for the active path, then finds the shortest lightpath that is link-disjoint from the candidate path. Computational studies show that the APF heuristic is quite effective in practice [6], but there is no performance guarantee for the APF heuristic. The ILP formulation of the problem can be used to find optimal solutions, but solving an ILP may take exponential time in the worst case. Thus ILPs are usually used only to solve problems known to be NP-Complete or problems with unknown complexity.

III. PROBLEM FORMULATION

A WDM network is modeled by an undirected graph $G = (V, E, \Lambda)$, where $V$ is the set of vertices, denoting nodes in the network; $E$ is the set of edges, denoting links (or optical fibers) in the network; $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_W\}$ is the set of wavelengths and $\Lambda(e) \subseteq \Lambda$ is the set of wavelengths available at link $e$.

We will use vertices and nodes interchangeably, as well as edges and links. We will use $n$ and $m$ to denote the number of nodes and links, respectively, and use $W$ to denote the total number of wavelengths a link may carry. Throughout this paper, we will assume that the network $G$ is connected. Therefore $m \geq n - 1$. In our model, each undirected link $(u, v)$ in the network represents a bidirectional link connecting $u$ and $v$. Whenever a link is used by a connection, it is occupied in both directions.

Definition 1: A lightpath $P(s, t, \lambda)$ between nodes $s \in V$ and $t \in V$ on wavelength $\lambda \in \Lambda$ is an $s$-$t$ path $\pi(s, t)$ in $G$ which uses wavelength $\lambda$ on every link of path $\pi(s, t)$. $\pi(s, t)$ is called the basepath of $P(s, t, \lambda)$.

Since each link has many wavelengths, several lightpaths may pass through the same link. The failure of a particular link $e$ may affect many existing connections—all the connections whose lightpaths use link $e$.

Link-Disjoint Path Pair with Wavelength Continuity Constraint (LDPPWCC)

Instance: A graph $G = (V, E, W, \Lambda)$, where $W$ represents the total number of wavelengths a link can carry, $\Lambda(e) \subseteq \Lambda = \{\lambda_1, \ldots, \lambda_W\}$ represents the set of available wavelengths on link $e \in E$; a source node $s$ and a destination node $t$.

Question: Is it possible to establish two link-disjoint paths from $s$ to $t$, such that both the paths satisfy the wavelength continuity constraint?

Note that $\Lambda(e)$ is a function of time as the set of available wavelengths at a particular link varies with time. In this paper, we are interested in the establishment of a new connection at the time instance $T$ when the request comes in. Therefore we assume that $\Lambda(e)$ is the set of available wavelengths on link $e$ at time $T$. 

...
Node-Disjoint Path Pair with Wavelength Continuity Constraint (NDPPWCC)

Instance: A graph \( G = (V, E, W, \Lambda) \), where \( W \) represents the total number of wavelengths a link can carry, \( \Lambda(e) \subseteq \Lambda = \{\lambda_1, \ldots, \lambda_W\} \) represents the set of available wavelengths on link \( e \in E \); a source node \( s \) and a destination node \( t \).

Question: Is it possible to establish two node-disjoint paths from \( s \) to \( t \), such that both the paths satisfy the wavelength continuity constraint?

Since our objective is to establish only two paths, and each path is required to use the same wavelength throughout, at most two wavelengths will be needed for path establishment. When both the paths can be established using the same wavelength, the problem reduces to finding two disjoint paths in a graph, polynomial time solutions for which are already well known [27], [28]. Accordingly, in this paper we concentrate on the scenario where both the paths cannot be established using the same wavelength. Instead of considering all \( W \) wavelengths, we will only consider two different wavelengths at a time. We will use \( \lambda_R \) and \( \lambda_B \) to denote the two different wavelengths under consideration. This leads to the following two problems.

Link-Disjoint Path Problem with Wavelength Continuity Constraint and 2 Wavelengths (LDPPWCC2W)

Instance: A graph \( G = (V, E, \Lambda) \), associated with each edge \( e \in E \), is \( \Lambda(e) \subseteq \{\lambda_R, \lambda_B\} \), indicating the set of free wavelengths (channels) associated with link \( e \); a source node \( s \) and a destination node \( t \).

Question: Is it possible to establish two link-disjoint paths from \( s \) to \( t \), such that one of the paths uses the wavelength \( \lambda_R \) and the other uses \( \lambda_B \)?

Node-Disjoint Path Problem with Wavelength Continuity Constraint and 2 Wavelengths (NDPPWCC2W)

Instance: A graph \( G = (V, E, \Lambda) \), associated with each edge \( e \in E \), is \( \Lambda(e) \subseteq \{\lambda_R, \lambda_B\} \), indicating the set of free wavelengths (channels) associated with link \( e \); a source node \( s \) and a destination node \( t \).

Question: Is it possible to establish two node-disjoint paths from \( s \) to \( t \), such that one of the paths uses the wavelength \( \lambda_R \) and the other uses \( \lambda_B \)?

The following lemma shows that the simplification from \( W \) wavelengths to 2 wavelengths does not lose any generality.

**Lemma 1:** LDPPWCC can be solved in polynomial time if and only if LDPPWCC2W can be solved in polynomial time. NDPPWCC can be solved in polynomial time if and only if NDPPWCC2W can be solved in polynomial time.

**Proof.** LDPPWCC2W is a special case of LDPPWCC. Therefore LDPPWCC can be solved in polynomial time only if LDPPWCC2W can be solved in polynomial time.

Now assume that LDPPWCC2W can be solved in polynomial time. To solve LDPPWCC, we can use Suurballe’s algorithm to see if the two link-disjoint paths can be established using the same wavelength \( \lambda \), for some \( \lambda \in \Lambda \). This can be done in polynomial time since there are at most \( W \) wavelengths and Suurballe’s algorithm runs in polynomial time. If such a pair of paths can be found in this way, we have a positive answer to LDPPWCC. If such a pair of paths cannot be found in this way, then there does not exist a pair of link-disjoint paths both using the same wavelength. Next, we can solve \( W(W - 1)/2 \) instances of LDPPWCC2W, using all combinations of \( \lambda_R \) and \( \lambda_B \) with \( \lambda_R, \lambda_B \in \Lambda \). This process will also take polynomial time, if LDPPWCC2W can be solved in polynomial time. There exists a pair of link-disjoint paths if and only if we can find such a pair for one of the \( W(W - 1)/2 \) LDPPWCC2W instances. Therefore if LDPPWCC can be solved in polynomial time if LDPPWCC2W can be solved in polynomial time. We can prove for the node version of the problems similarly.

LDPPWCC2W (NDPPWCC2W, respectively) can be viewed as a path finding problem where each link is colored using either red, green or blue, where the colors red, green and blue indicate that wavelength \( \lambda_R, \lambda_B \), and both, respectively, are available for path establishment on that link. We want to find two link-disjoint (node-disjoint, respectively) paths from node \( s \) to node \( t \) such that one path (called the red path) uses only red and green links and the other (called the blue path) uses only blue and green links. Formally, the problem can be stated as follows:

**Link Disjoint Path Problem in Graphs with Red, Green, Blue Links (LDPRGB)**

Instance: A graph \( G = (V, E) \), where each link \( e \in E \) is colored red, blue or green; a source node \( s \) and a destination node \( t \).

Question: Is it possible to establish two link-disjoint paths from \( s \) to \( t \), such that one of the paths uses only the red and green links and the other uses the blue and green links?

**Node Disjoint Path Problem in Graphs with Red, Green, Blue Links (NDPRGB)**

Instance: A graph \( G = (V, E) \), where each link \( e \in E \) is colored red, blue or green; a source node \( s \) and a destination node \( t \).

Question: Is it possible to establish two node-disjoint paths from \( s \) to \( t \), such that one of the paths uses only the red and green links and the other uses the blue and green links?
Note that LDPPRGB is equivalent to LDPPWCC2W and NDPPRGB is equivalent to NDPPWCC2W. In the following section we will prove that both LDPPRGB and NDPPRGB are NP-Complete. Therefore the problems LDPPWCC2W and NDPPWCC2W are also NP-Complete. This in turn implies that the problems LDPPWCC and NDPPWCC are NP-Complete.

IV. Complexity Analysis

We will consider a graph whose edges are colored by one of the three colors: red, green and blue. Recall that a red path is a path whose links are either red or green, and a blue path is a path whose links are either blue or green.

Theorem 2: The problem LDPPRGB is NP-Complete.

Proof. Clearly LDPPRGB is in NP, as one can verify efficiently whether two paths are link-disjoint. In the rest, we will prove that the problem is also NP-hard.

We give a polynomial time reduction from 3SAT to LDPPRGB. Since 3SAT is NP-hard, this will prove the theorem. The reduction maps a 3-CNF formula $\phi$ (an instance of the 3SAT problem) to an LDPPRGB instance $(G, s, t)$ so that $\phi$ is satisfiable if and only if $G$ contains a red $s$–$t$ path $p^R$ and a blue $s$–$t$ path $p^B$ such that $p^R$ and $p^B$ are link-disjoint in $G$.

We describe the mapping in two steps, while illustrating the process with the sample 3SAT instance $\phi = (\bar{x}_1 \lor \bar{x}_2 \lor x_3) \land (\bar{x}_1 \lor x_2 \lor x_3) \land (x_1 \lor x_2 \lor x_3) \land (\bar{x}_1 \lor x_2 \lor \bar{x}_3)$.

Step 1. Let $L = \{x_1, \bar{x}_1, \ldots, x_l, \bar{x}_l\}$ be the set of number of variables and use $k$ to denote the number of clauses.

We define a graph $G_1$ in the following way: Create vertices $s$ and $t$. For each clause-literal pair $(C_i, x_j)$, create two vertices $u_{ij0}$ and $v_{ij1}$ and a green edge connecting them. For each clause-literal pair $(C_i, \bar{x}_j)$, create two vertices $u_{ij0}$ and $v_{ij1}$ and a green edge connecting them. For each variable $x_j$, create two vertices $u_j$ and $v_j$ and red edges $(u_j, u_{ij1})$, $(u_j, v_{ij0})$, $(v_j, u_{ij1})$, $(v_j, v_{ij0})$, as well as red edges $(u_{ij1}, u_{i+1j1})$ and $(v_{ij0}, u_{i+1j0})$ for $i = 1, 2, \ldots, k - 1$. Finally we add red edges $(s, u_1)$, $(u_l, t)$ and red edges $(v_1, u_{j+1})$ for $j = 1, 2, \ldots, l - 1$. The graph $G_1$ corresponding to the sample formula is shown in Fig. 2.

From the construction of $G_1$ and Fig. 2 we can see the following facts: There are $2^l$ different red $s$–$t$ paths in $G_1$. Let $p_R$ be any red $s$–$t$ path. For any variable $x_j$, $p_R$ either goes through all of the vertices corresponding to literal $x_j$ and none of the vertices corresponding to literal $\bar{x}_j$, or goes through all of the vertices corresponding to literal $\bar{x}_j$ and none of the vertices corresponding to literal $x_j$.

Step 2. Next we add $k + 1$ vertices and $6k + 2$ blue edges to $G_1$ in the following way to obtain graph $G_2$: Add vertices $w_0$, $w_1$, $\ldots$, $w_k$ and blue edges $(s, w_0)$ and $(w_k, t)$.

Let $C_i$ be the $i$th clause and $\alpha_i$ be one of the three literals in $C_i$. If $\alpha_i$ is $x_j$, we add two blue edges $(w_{i-1}, u_{ij1})$ and $(w_1, v_{ij1})$; if $\alpha_i$ is $\bar{x}_j$, we add two blue edges $(w_{i-1}, u_{ij0})$ and $(w_1, v_{ij0})$. Therefore $6k$ blue edges are added in this way. The graph $G_2$ corresponding to the sample formula is shown in Fig. 3.

From the construction of $G_2$ as shown in Fig. 3 we can see the following facts: There are $k^l$ different blue $s$–$t$ paths in $G_2$. Let $p^B$ be any red $s$–$t$ path in $G_2$ and $p^B$ be any blue $s$–$t$ path in $G_2$ which is link-disjoint with $p^R$. For any variable $x_j$, if $p^B$ goes through any of the vertices corresponding to literal $x_j$, then $p^B$ must go through all of the vertices corresponding to literal $\bar{x}_j$. Similarly, if $p^B$ goes through any of the vertices corresponding to literal $\bar{x}_j$, then $p^B$ must go through all of the vertices corresponding to literal $x_j$.

It is clear that graph $G_2$ can be constructed from formula $\phi$ in polynomial time, since $G_2$ contains $4kl + 2l + k + 3$ nodes, $2kl + 3l + 1$ red edges, $2kl$ green edges and $6k + 2$ blue edges.

We now prove the desired result that $(G_2, s, t)$ contains a pair of link-disjoint red and blue $s$–$t$ paths if and only if $\phi$ is satisfiable. Let us first assume that $(G_2, s, t)$ contains link-disjoint paths $p^R$ and $p^B$, where $p^R$ goes through only red and green edges and that $p^B$ goes...
through only blue and green edges. We will define a truth assignment of the variables \( f: \{x_1, x_2, \ldots, x_l\} \mapsto \{TRUE, FALSE\} \) so that \( \phi \) is true under this assignment.

Let \( x_j \) be any literal in \( \phi \). If \( p^B \) goes through any of the vertices corresponding to literal \( x_j \), then \( p^R \) must go through all of the vertices corresponding to literal \( \bar{x}_j \). In this case, we assign \( f(x_j) = TRUE \).

If \( p^B \) goes through any of the vertices corresponding to literal \( \bar{x}_j \), then \( p^R \) must go through all of the vertices corresponding to literal \( x_j \). In this case, we assign \( f(x_j) = FALSE \).

If \( p^B \) does not go through any of the vertices corresponding to literal \( x_j \) or literal \( \bar{x}_j \), then \( p^R \) goes through either all of the vertices corresponding to literal \( x_j \) or all the vertices corresponding to literal \( \bar{x}_j \). In the former case, we set \( f(x_j) = FALSE \). In the latter case, we set \( f(x_j) = TRUE \). In short, \( p^B \) goes through the vertices \( w_i \) and the edges \( (u_{ij1}, v_{ij1}) \) (with \( f(x_j) = TRUE \)) and edges \( (u_{ij0}, v_{ij0}) \) (with \( f(x_j) = FALSE \)) in the truth assignment. Since the blue path \( p^B \) goes through the vertices \( w_0, w_1, \ldots, w_c \) in that order, \( \phi \) is satisfied with the above truth assignment.

Fig. 4 illustrates the truth assignment corresponding to the blue path \( p^B \): \( s-w_0-u_{110}-v_{110}-w_1-u_{210}-v_{210}-w_2-u_{321}-v_{321}-w_3-u_{421}-v_{421}-w_4-t \) and the red path \( p^R \): \( s-u_{111}-v_{111}-u_{211}-v_{211}-u_{311}-v_{311}-u_{411}-v_{411}-u_{62}-u_{120}-u_{220}-v_{220}-u_{320}-v_{320}-u_{420}-v_{420}-u_{53}-u_{131}-v_{131}-u_{231}-v_{231}-u_{331}-v_{331}-u_{431}-v_{431}-c-l \). Since \( p^B \) goes through the vertices \( u_{110}, v_{110}, u_{210}, v_{210} \) which correspond to the literal \( \bar{x}_1 \), we set \( f(x_1) = FALSE \) in the truth assignment. Since \( p^B \) goes through the vertices \( u_{321}, v_{321}, u_{421}, v_{421} \) which correspond to the literal \( x_2 \), we set \( f(x_2) = TRUE \) in the truth assignment. For this example, \( p^B \) does not go through any of the vertices corresponding to the literals \( x_3 \) and \( \bar{x}_3 \). We can assign \( f(x_3) \) as do not care in the truth assignment. Since \( p^R \) goes through the vertices corresponding to the literal \( x_3 \), we set \( f(x_3) = FALSE \) in the truth assignment. One can easily verify that \( \phi \) evaluates to \( TRUE \) with this truth assignment.

To show the converse, assume \( \phi \) is satisfiable and let \( f \) be a truth assignment that satisfies \( \phi \). We will show that there exist a red \( s-t \) path \( p^R \) and a blue \( s-t \) path \( p^B \) which are link-disjoint in \( G_2 \).

Note that any red \( s-t \) path will contain the edges \( (s, u_j), (v_j, t) \) for \( j = 1, 2, \ldots, l \), and a red path from \( u_j \) to \( v_j \) for \( j = 1, 2, \ldots, l \). If \( f(x_1) = FALSE \), we define the segment of \( p^R \) from \( u_j-v_j \) to be \( s-u_1-u_{111}-v_{111}-u_{211}-v_{211}-\cdots-u_{c11}-v_{c11}-v_1 \). Otherwise, we define the segment of \( p^R \) from \( u_j-v_j \) to be \( s-u_1-u_{110}-v_{110}-u_{210}-v_{210}-\cdots-u_{c10}-v_{c10}-t \). We define the blue \( s-t \) path \( p^B \) to contain the blue edges \( (s, w_0), (w_c, t) \) and one green edge and two blue edges for each clause in the sample 3SAT instance \( \phi \). Let \( C_i \) be the \( i \)th clause. Since \( C_i \) is \( TRUE \) under truth assignment \( f \), at least one of the three literals of \( C_i \) is \( TRUE \). Let \( x_j \) be one such literal. The blue path \( p^B \) contains the blue edges \( (w_{i-1}, u_{ij1}), (v_{ij1}, w_i) \) and the green edge...
(uij1, vij1) (the blue edges (w1−1, uij0), (vij0, wi) and the green edge (uij0, vij0), respectively).

Consider the sample the sample 3SAT instance \( \phi \) again. The truth assignment \( f(x_1) = \text{TRUE}, f(x_2) = \text{TRUE}, f(x_3) = \text{FALSE} \) satisfies \( \phi \). Refer to \( G_2 \) in Fig. 3, the corresponding blue path is \( p^B: s-u_0-v_130-w_1-u_221-v_221-u_2-311-v_311-w_3-u_421-v_421-w_4-t \), and the corresponding red path is \( p^R: s-u_1-v_110-w_120-v_220-u_320-v_320-u_420-v_420-w_2-u_131-v_131-u_231-v_231-u_331-v_331-u_431-v_431-w_4-t \).

Since \( p^B \) does not go through any vertices corresponding to literal whose value is \( \text{FALSE} \) and the \( u_i-v_i \) segment of \( p^R \) goes through only those vertices corresponding to a literal whose value is \( \text{not TRUE} \), \( p^B \) and \( p^R \) are link-disjoint. This proves that LDPPRGB is NP-hard. Since LDPPRGB is both in NP and NP-hard, it is NP-Complete.

**Theorem 3:** The problem NDPPRGB is NP-Complete.

**Proof:** Again we prove this using reduction from 3SAT. Given a 2-CNF formula \( \phi \), we construct the same graph \( G_2 \) as in the proof of Theorem 2. It is clear that the graph construction such that a red \( s-t \) path \( p^R \) and a blue \( s-t \) path \( p^B \) are link-disjoint if and only if they are node-disjoint. It then follows the argument of the proof of Theorem 2 that NDPPRGB is also NP-hard. It is easy to verify whether two paths are node-disjoint. Therefore NDPPRGB is NP-Complete.

Given that the problem is NP-Complete, it is appropriate to study heuristic algorithms and ILP solutions. In the rest of this paper, we will concentrate on LDPPWCC and study both heuristic algorithms and ILP based optimal solutions.

**V. APPROXIMATE SOLUTION USING HEURISTIC TECHNIQUE**

We consider the situation that some active-backup paths are in existence while a new connection request with source node \( s \) and destination node \( t \) arrives. Recall that each link can carry up to \( W \) channels \( (W \) different wavelengths). A channel is free if it is not used by an active or a backup path. A channel is active if it is used by an active path. A channel is reserved if it is used by a backup path. We assume that we have full knowledge of the network status.

The commonly used active path first (APF) heuristic is described in the following.

The advantage of heuristic APF is that it is simple and runs fast. If the active path \( AP \) is chosen correctly, the heuristic will find the link-disjoint path pair. However, there are cases where the heuristic APF may fail to find a pair of link-disjoint paths, even when such a pair exists. Fig. 5 shows an example for which APF fails.

![Fig. 5. An example for which APF fails.](image_url)

**Algorithm 1 APF (G, s, t)**

**step.1** Find an \( s-t \) lightpath \( AP \) on free channels using the minimum number of links.

**step.2** Let \( G' \) be a copy of \( G \). Remove every active or reserved channel. Remove every free channel using a link on \( AP \).

**step.3** Find an \( s-t \) lightpath \( BP \) in \( G' \) using the minimum number of links.

**step.4** IF both \( AP \) and \( BP \) can be found, output the two paths. Otherwise, the request is blocked.

In the example shown in Fig. 5, APF will find the path \( s-x-w-t \) (on \( \lambda_1 \)) as the candidate for the active path \( AP \). However, after removing the links along this candidate active path, \( s \) and \( t \) is no longer connected in the network. Therefore the heuristic APF cannot find a pair of link-disjoint \( s-t \) paths. Clearly, we can use \( s-u-v-w-t \) (on \( \lambda_2 \)) as the active path and use \( s-x-y-z-t \) (on \( \lambda_1 \)) as the backup path. In the following, we will present an enhanced active path first heuristic (called APFE) which can avoid pitfalls like this. In the description of APFE, \( M \) represents a very big number which is greater than any integer and \( i \cdot M > j \cdot M \) for integers \( i > j \).

We note that the heuristic APFE will find a pair of link-disjoint paths whenever the heuristic APF can find such a pair. However, APFE may find a link-disjoint path pair even when APF fails to do so. APFE tries to reduce the number of shared links in the pair of active-backup paths iteratively. When that number cannot be reduced during one iteration, the heuristic stops and blocks the connection request. We note that the heuristic APFE is not guaranteed to find a pair of link-disjoint paths when one exists.

For the example network shown in Fig. 5, APFE will first find the path \( s-x-w-t \) (on \( \lambda_1 \)) as the candidate for the active path \( AP \). It then find the path \( s-u-v-w-t \) (on \( \lambda_2 \)) as a candidate for the backup path \( BP \). Note that \( AP \) and
Algorithm 2 APFE (G, S, T)

**step 1** Find an \(s\-t\) lightpath \(AP\) on free channels using the minimum number of links. Set \(\text{cost} = \infty\).

**step 2** Let \(G'\) be a copy of \(G\). For every active or reserved channel, assign a cost of \(\infty\). For every free channel on a link on \(AP\), assign a cost of \(M\). For every other free channel, assign a cost of 1.

**step 3** Find a minimum cost \(s\-t\) lightpath \(BP\) in \(G'\).

**step 4** IF \(AP\) and \(BP\) are link-disjoint THEN
\[ AP \text{ and } BP \text{ are the active and backup paths for the connection request; STOP} \]
ELSEIF the cost \(BP\) is at least \(\text{cost}\) THEN
\[ \text{STOP failure} \]
ELSE
\[ \text{Set } \text{cost} \text{ to the cost of } BP \text{ and let } AP \text{ represent } BP. \text{ goto Step 3.} \]
ENDIF

\(BP\) shares a common link \((w, t)\). The heuristic then discards \(AP\) and finds \(s\-x\-y\-z\-t\) (on \(\lambda_1\)) as the active path \(AP\). This time, \(AP\) and \(BP\) are link-disjoint so APFE finds the pair of link-disjoint paths successfully. However, there are examples for which APFE fails. For performance studies, we present an integer linear programming (ILP) formulation in the next section and present simulation results comparing APF, APFE and ILP on sample network topologies.

VI. EXACT SOLUTION USING ILP

To evaluate the performance of the heuristic algorithms, we formulate the following ILP for LDPPWCC2W with wavelengths \(\lambda_R\) and \(\lambda_B\) and source-destination node pair \(s\) and \(t\).

For each undirected link \((u, v) \in E\), define \(R(u, v) = 1\) if \(\lambda_R\) is available on \((u, v)\) and \(R(u, v) = 0\) otherwise. Similarly, define \(B(u, v) = 1\) if \(\lambda_B\) is available on \((u, v)\) and \(B(u, v) = 0\) otherwise. Define a function \(\delta_{st}(\cdot)\) on \(V\) such that \(\delta_{st}(s) = 1\), \(\delta_{st}(t) = -1\), \(\delta_{st}(v) = 0\) for every other node \(v \in V\). For each undirected link \((u, v) \in E\), we associate four 0/1 variables \(r(u, v), b(u, v), r(v, u), b(v, u)\). Note that \(r(u, v)\) and \(r(v, u)\) are different although \((u, v)\) and \((v, u)\) denote the same undirected edge. The same can be said about \(b(u, v)\) and \(b(v, u)\). We will use the \(r(u, v)\) variables to define a red path from \(s\) to \(t\) and use the \(b(u, v)\) variables to define a blue path from \(s\) to \(t\). The integer linear programming formulation is given in the following:

\[
(P): \quad \min \sum_{[u, v] \in E} r(u, v) + r(v, u) + b(u, v) + b(v, u) \\
\text{s.t. } \sum_{v \in V} r(x, v) - \sum_{u \in V} r(u, x) = \delta_{st}(x), \quad \forall x \in V, \\
\sum_{v \in V} b(x, v) - \sum_{u \in V} b(u, x) = \delta_{st}(x), \quad \forall x \in V, \\
r(u, v) + r(v, u) \leq R(u, v), \quad \forall (u, v) \in E, \\
b(u, v) + b(v, u) \leq B(u, v), \quad \forall (u, v) \in E, \\
r(u, v) + b(u, v) \leq 1, \quad \forall (u, v) \in E, \\
r(u, v), b(u, v), b(v, u) \geq 0, \quad \forall (u, v) \in E.
\]

The following is an explanation of the ILP formulation (P). Recall that \(R(u, v) = 1\) if and only if wavelength \(\lambda_R\) is available on link \((i, j)\) and \(B(u, v) = 1\) if and only if wavelength \(\lambda_B\) is available on link \((i, j)\). Therefore a feasible solution to (P) corresponds to a red \(s\-t\) lightpath on wavelength \(\lambda_R\), defined by the nonzero variables \(r(u, v)\), and a blue \(s\-t\) lightpath on wavelength \(\lambda_B\), defined by the nonzero variables \(b(u, v)\).

The constraint \(r(u, v) + r(v, u) \leq R(u, v)\) ensures that the red path can only use the links on which wavelength \(\lambda_R\) is available. The constraint \(b(u, v) + b(v, u) \leq B(u, v)\) ensures that the blue path can only use the links on which wavelength \(\lambda_B\) is available. The constraint \(r(u, v) + b(u, v) + b(v, u) \leq 1\) ensures that the red path and the blue path are link-disjoint.

Note that we have to solve up to \(W \cdot (W - 1)\) instances of (P) to solve problem LDPPWCC, since we need to loop over all possible wavelength combinations. The cases where \(\lambda_R = \lambda_B\) can be solved using the polynomial time algorithm of Saurballe [27], [28].

We choose to use this ILP formulation because (1) it is simpler to understand and to implement since only two wavelengths are involved; (2) the smaller the ILP, the easier to solve, as ILPs may require exponential time to solve.

VII. RESULTS AND DISCUSSION

To evaluate the effectiveness of the heuristics and ILP based algorithms, we have compared them using the 20-node Arpanet topology (see Fig. 6) and the 33-node Italian National Network topology (see Fig. 7). All three algorithms are implemented on a SUN Ultra machine using programming the language C and the optimization package CPLEX 6.5.

For each network topology, we construct 9 WDM networks, obtained by having 5, 10 and 20 wavelengths per...
TABLE I
SIMULATION RESULTS ON ARPA-NET

<table>
<thead>
<tr>
<th>Wavelengths per Link</th>
<th>Network Load(%)</th>
<th>Average Time (in milliseconds)</th>
<th>Statistics for APF, APFE and ILP (exact)</th>
<th>Total number of Cases</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>APF</td>
<td>APFE</td>
<td>ILP(exact)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>YYY</td>
<td>NYY</td>
<td>NNY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>25</td>
<td>0.632</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>50</td>
<td>0.316</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>75</td>
<td>0.316</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>25</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>50</td>
<td>0.947</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>75</td>
<td>0.421</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20</td>
<td>50</td>
<td>1.053</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20</td>
<td>75</td>
<td>0.632</td>
</tr>
</tbody>
</table>

TABLE II
SIMULATION RESULTS ON ITALIAN NATIONAL NETWORK

<table>
<thead>
<tr>
<th>Wavelengths per Link</th>
<th>Network Load(%)</th>
<th>Average Time (in milliseconds)</th>
<th>Statistics for APF, APFE and ILP (exact)</th>
<th>Total number of Cases</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>APF</td>
<td>APFE</td>
<td>ILP(exact)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>YYY</td>
<td>NYY</td>
<td>NNY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>25</td>
<td>1.875</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>50</td>
<td>1.080</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>75</td>
<td>0.455</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>25</td>
<td>3.371</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>50</td>
<td>2.045</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>75</td>
<td>0.663</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20</td>
<td>25</td>
<td>5.947</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20</td>
<td>50</td>
<td>3.674</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20</td>
<td>75</td>
<td>1.042</td>
</tr>
</tbody>
</table>

Fig. 6. 20-Node ARPANET Topology.

link and network loads of 25%, 50% and 75%. By a network with 50% load, we mean that 50% of the channels in the network are marked as either active or reserved. Clearly, the higher the network load, the high the blocking probability. For each of the 18 WDM networks, we consider all possible source-destination node pairs and try to find a pair of link-disjoint source-destination lightpaths using the optimal algorithm and the two heuristics. For example, for the 9 WDM networks generated from the Arpanet topology, we consider all 190 = 20 * (20 − 1)/2 possible source-destination pairs as the connection requests.

For each connection request, there are four possible outcomes:

- NNN: meaning none of the three algorithms found a pair of link-disjoint paths;
- NNY: meaning both heuristics failed to find a pair of link-disjoint paths but the ILP exact algorithm found one;
- NYY: meaning the first heuristic failed to find a pair of link-disjoint paths but the second heuristic and the ILP exact algorithm both found one;
- YYY: meaning all three algorithms found a pair of link-disjoint paths.
Simulation results for networks corresponding to the 20-node Arpanet are shown in Table I. Simulation results for networks corresponding to the 33-node Italian National Network are shown in Table II.

As an example, the entries in the first row of Table I have the following meaning (from left to right): Maximum number of wavelengths per link is 5; The network load is 25%; The average time (in milliseconds) used by APF is 0.632; The average time (in milliseconds) used by APFE is 1.158; The average time (in milliseconds) used by ILP is 17.579; In 176 cases, all three algorithms were successful; In 11 cases, APFE and ILP were both successful while APF was not; In 2 cases, ILP was successful while APFE was not; In 1 case, none of the three algorithms can find a path pair, because there was no pair of link-disjoint paths; Total number of connection requests was 190.

From Table I, we can see that APFE found a pair of disjoint paths in 26 of the 28 cases for which APF failed. From Table II, we can see that APFE found a pair of disjoint paths in 73 of the 81 cases for which APF failed. This shows that APFE is noticeably better than APF, while having a similar running time.

One can see that APFE uses only a fraction of the time used by ILP. Among the 6462 test cases shown in the two tables, APFE failed to find a pair of paths when one existed only 10 times. In other words, APFE found optimal solutions in 99.8% of the cases, while using no more than 6 milliseconds in any particular case.

VIII. CONCLUSION

In this paper, we presented a rigorous proof that finding a pair of link-disjoint or node-disjoint paths with the wavelength continuity constraint in a WDM network is NP-Complete. To the best of our knowledge, this is the first rigorous proof for this problem. We then presented an enhanced active path first heuristic and an ILP based algorithm. Computational results show that our enhanced active path first heuristic outperforms the commonly used active path first heuristic and finds optimal solutions in almost all cases, while spending only a fraction of the time used by the ILP based algorithm. Since finding a path with dedicated protection is NP-Complete, we believe that the more complicated problem of finding a path with shared protection is also NP-Complete. Currently we are investigating the computational complexity of shared backup path provisioning and effective heuristics for that problem.

ACKNOWLEDGMENT

We would like to acknowledge the assistance of Bao Shen for the ILP implementations and Bin Hao and Rakesh Banka for the heuristic implementations. The research of Guoliang Xue was supported in part by NSF ITR grant ANI-0312635 and ARO grant DAAD19-00-1-0377.

REFERENCES

In this section, we will show that the problem of computing a pair of link-disjoint (or node-disjoint) active-backup paths without the wavelength continuity constraint in a WDM network with wavelength converters [24] at the network nodes is computationally easy.

Let $G = (V, E, \Lambda)$ be the WDM network with $n$ nodes, $m$ links, and a maximum of $W$ wavelengths per link. Define the wavelength union graph $\overline{G}$ of $G$ by deleting every link in $G$ for which there is no available wavelengths. The wavelength union graph of the network in Fig. 8 is shown in Fig. 9. Basically, a link of $G$ is a link $\overline{G}$ if the link has at least one available wavelength.

Clearly $\overline{G}$ can be constructed from $G$ in $O(Wm)$ time. We can then use Suurballe’s algorithm [27], [28] to compute a shortest pair of link-disjoint (or node-disjoint) paths connecting $s$ and $t$ (or to confirm the nonexistence of such a pair) in $\overline{G}$. If such a pair does not exist, then clearly there is no link-disjoint (node-disjoint) path pairs connecting $s$ and $t$ in $G$. Let $\pi_1$ and $\pi_2$ be the shortest pair of link-disjoint (or node-disjoint) $s$–$t$ paths. For each link on $\pi_1$ or $\pi_2$, we can assign an available wavelength. Therefore $\pi_1$ and $\pi_2$ can be used as a pair of active-backup paths in a WDM network with wavelength converters at the nodes.

**Appendix**

In this section, we will show that the problem of computing a pair of link-disjoint (or node-disjoint) active-backup paths without the wavelength continuity constraint in a WDM network with wavelength converters [24] at the network nodes is computationally easy.

Let $G = (V, E, \Lambda)$ be the WDM network with $n$ nodes, $m$ links, and a maximum of $W$ wavelengths per link. Define the wavelength union graph $\overline{G}$ of $G$ by deleting every link in $G$ for which there is no available wavelengths. The wavelength union graph of the network in Fig. 8 is shown in Fig. 9. Basically, a link of $G$ is a link $\overline{G}$ if the link has at least one available wavelength.

Clearly $\overline{G}$ can be constructed from $G$ in $O(Wm)$ time. We can then use Suurballe’s algorithm [27], [28] to compute a shortest pair of link-disjoint (or node-disjoint) paths connecting $s$ and $t$ (or to confirm the nonexistence of such a pair) in $\overline{G}$. If such a pair does not exist, then clearly there is no link-disjoint (node-disjoint) path pairs connecting $s$ and $t$ in $G$. Let $\pi_1$ and $\pi_2$ be the shortest pair of link-disjoint (or node-disjoint) $s$–$t$ paths. For each link on $\pi_1$ or $\pi_2$, we can assign an available wavelength. Therefore $\pi_1$ and $\pi_2$ can be used as a pair of active-backup paths in a WDM network with wavelength converters at the nodes.