J Math Model Algor (2013) 12:407–428
DOI 10.1007/s10852-013-9221-0

Designing WDM Optical Networks Using Branch-and-Price

S. Raghavan · Daliborka Stanojević

Received: 4 April 2011 / Accepted: 15 February 2013 / Published online: 8 March 2013

Abstract In this paper, we present an exact solution procedure for the design of two-layer wavelength division multiplexing (WDM) optical networks with wavelength changers and bifurcated flows. This design problem closely resembles the traditional multicommodity flow problem, except that in the case of WDM optical networks, we are concerned with the routing of multiple commodities in *two* network layers. Consequently, the corresponding optimization models have to deal with two types of multicommodity variables defined for each of the network layers. The proposed procedure represents one of the first branch-and-price algorithms for a general WDM optical network setting with no assumptions on the number of logical links that can be established between nodes in the network. We apply our procedure in a computational study with four different network configurations. Our results show that for the three tested network configurations our branch-and-price algorithm provides solutions that are on average less than 5 % from optimality. We also provide a comparison of our branch-and-price algorithm with two simple variants of the upper bounding heuristic procedure HLDA that is commonly used for WDM optical network design.

Keywords WDM optical networks · Branch-and-price · Network flows · Multilayer network design

S. Raghavan (✉)
Smith School of Business & Institute for Systems Research,
University of Maryland, College Park, MD 20742, USA
e-mail: raghavan@umd.edu

D. Stanojević
School of Medicine, University of Maryland,
Baltimore, MD 21201, USA
e-mail: dstanojevic@umm.edu

1 Introduction

Optical networks with wavelength division multiplexing (WDM) belong to the second generation of optical networks, designed to take advantage of the large optical network bandwidth. One of the main distinguishing characteristics of WDM optical networks is their multi-layer nature, which requires the simultaneous routing of both the traffic and the lightpaths (which are specific logical paths established over the physical topology) in the corresponding network layers. The two layers commonly considered in WDM optical network design include the physical topology defined by the physical network of optical fibers, and the virtual (logical) topology defined by lightpaths established over the physical topology.

The WDM optical network design problem in this network setting seeks to determine:

i. the optimal number and routing of the lightpaths in the physical topology, and
ii. routing of the traffic over the logical topology (in the logical topology each lightpath can be seen as an edge that can be used for routing of traffic)

so that the desired objective is minimized/maximized.

Depending on the type of the optical network considered, this problem may require the consideration of various restrictions on the physical and logical topology. The restrictions related to the physical topology of WDM optical networks usually depend on the type of node equipment used and the capacity of the optical fibers available in the network.

The WDM optical network design problem studied in this paper deals with an all-optical (also referred to as transparent) architecture. Current trends in optical networks indicate they are gradually moving from an opaque architecture (i.e., an architecture where the optical signal carrying traffic undergoes an optical-electronic-optical (OEO) conversion at each node) to a transparent one. There are several design choices in the deployment of transparent optical networks. First, in WDM networks a lightpath is required to use the same wavelength (this is referred to as a wavelength continuity constraint in the literature) across the entire path unless intermediate nodes are equipped with wavelength convertors. Currently, commercially available wavelength convertor technology uses OEO conversion. Thus, currently transparent optical networks must explicitly address the wavelength continuity constraint. Alternatively, wavelength convertors are placed selectively at some of the nodes in the network (such an architecture where OEO conversion only takes place at a subset of nodes is called translucent) allowing the wavelength continuity constraint to be relaxed somewhat. One advantage of networks with wavelength convertors (pointed out by [26]) is some modest improvement in network throughput. Currently, there is considerable focus on the development of optical wavelength converters (see [1, 11, 12, 24, 28] for example). All optical devices allow for the design of optical networks with lower power consumption (which is a very important area of research). Thus, some advantages of a transparent optical architecture with optical wavelength convertors are (i) significantly lower power consumption, and (ii) increased network throughput. Our paper focuses on transparent optical networks and assumes the availability of optical wavelength convertors. Thus the wavelength continuity constraint can be relaxed.

In considering the restrictions imposed by the capacity of an optical fiber, the number of wavelengths that can be supported by a given fiber is typically considered. Finally, an important issue related to the physical topology is whether the network already exists, has a possibility of expanding, or is just being built. Although some researchers take network expansion into consideration (see for example [5]), it is often assumed that either the physical topology is given and fixed [19, 22], or, that only a part of the physical topology is fixed, and the rest is subject to change. (For example, Hu and Leida [15] and Konda and Chou [17] assume that the number of transmitters and receivers available at the network nodes is unknown and needs to be determined, while the remaining part of the physical topology is fixed.)

In this paper, we consider the following situation faced by a telecommunications service provider. Here the physical topology is given and fixed. The service provider seeks to *maximize* the amount of traffic routed over its network. (Since any traffic that is not routed on its own network can be transported on a competitors network at a cost, it is desirable to route as much of the demand as possible on one's own network.)

The logical topology requirements related to WDM optical network design are concerned with decisions on which nodes should be connected by the lightpaths, and how these lightpaths should be routed in the physical layer. These requirements may include restrictions on the number of physical hops (actual physical links) used by each lightpath, and propagation delays associated with a specific lightpath routing. Additionally, the logical topology can be designed with symmetrical lightpaths throughout the network (that is, for each specific lightpath from node i to node j, there should also be a corresponding, symmetrical lightpath from node j to node i). However, given that asymmetric traffic requests tend to be more efficiently served using asymmetrical lightpaths, we assume that physical links in the network are unidirectional, and that the symmetry of lightpaths is not required.

Finally, an important requirement related to the design of WDM optical networks is whether bifurcation of traffic should be allowed or not (that is whether traffic of a single commodity can be sent over multiple lightpaths or not). As indicated by Ramaswami and Sivarajan [23], bifurcation of traffic is not a problem if the traffic is represented by IP packets and we, also, make an assumption that the bifurcation of traffic is allowed in the network. We should note, however, that bifurcation increases the complexity and the cost of traffic reassembly, and may also introduce delay jitter at the destination of the traffic. In a related paper [21], we consider and address the situation where bifurcation of flow is not permitted (which is more appropriate for SONET circuits and networks that allow grooming[1] in order to improve the usage of bandwidth and the components of the higher speed connections).

The general WDM optical network design problem is often seen as three interrelated problems: the logical topology design problem, the traffic routing problem, and the lightpath routing problem. The logical topology design problem is defined as the problem that is concerned with determining the number of lightpaths to be

[1]The issue of traffic grooming is relevant and especially important in situations where bifurcation of flow is not permitted. Since traffic cannot be split, a more efficient usage of bandwidth may be obtained by judiciously combining demands/groomingthe traffic. The paper by Cinkler et al. [8] provides a nice overview of the topic and associated results.

established between all pairs of nodes. The traffic routing problem deals with the routing of the traffic over the established logical topology. By this definition, the logical topology design problem and the traffic routing problem are not concerned with the actual routing of the lightpaths in the associated physical layer. Instead, the part of the WDM optical network design problem that deals with the routing of lightpaths in the physical topology (given the number of lightpaths that need to be established between all pairs of nodes) is typically known as the lightpath routing and wavelength assignment (RWA) problem.

In this paper, because we work in the realm of optical wavelength convertors, we use slightly different terminology for the subproblems of the WDM optical network design problem (note however, the terminology used is consistent with [22]). Specifically, we use the term **logical topology design** (LTD) for the problem that completely defines the logical topology (both the number and the routing of lightpaths in the physical topology), and the term **traffic routing** for the problem of traffic routing over the logical topology.

The rest of the paper is organized as follows. Section 2 presents a discussion of some of the relevant literature, and places the contributions of our work in that context. Section 3 presents the mathematical formulation for the problem studied in this paper. In Section 4, we propose a novel branch-and-price algorithm for the formulation presented in Section 3. In Section 5, we propose two simple heuristic procedures that can be used to obtain fast upper bounds for the WDM optical network design problem. Section 6 describes a computational study of the branch-and-price procedure. The results indicate that our branch-and-price procedure provides solutions that are on average within 2.5 % of optimality on networks with up to 20 nodes. Section 7 provides concluding remarks.

2 Literature Review

There are several excellent monographs that discuss WDM optical network design. See for example [4, 18, 23, 25]. Most of the previous research has focused on subproblems of the WDM optical network design problem or has solved the WDM optical network design problem by solving the subproblems sequentially. For example, several heuristic procedures usually simplify the LTD problem by either using a heuristic to define the logical topology, or by pre-specifying a small subset of lightpaths that can be used for the logical topology design. Once the LTD problem is solved, the traffic routing problem over a fixed logical topology can be solved as a standard origin-destination multicommodity flow (ODMCF) problem. Ramaswami and Sivarajan, have, for example, used this approach in [22], where they proposed several heuristic procedures for the LTD problem. The logical topology determined this way was then used as input for a mixed-integer program (MIP) that solves the traffic routing problem over the fixed logical topology. The computational results in [22] indicate that the linear program based on the logical topology determined by one of the LTD procedures called HLDA (an acronym for heuristic logical topology design algorithm) provides the least congestion in most cases when compared to other heuristic LTD procedures proposed in the same paper.

Haque et al. [14] used a column generation based approach for the traffic routing over a fixed logical topology (i.e., their method ignores the design of the logical topology and assumes it is given). Their procedure is part of an algorithm that solves the WDM optical network design problem in optical networks that (i) do not have wavelength conversion abilities (so they solve the RWA problem), (ii) impose a constraint on maximum allowable delay for each commodity, and (iii) allow at most one lightpath between any two nodes. Computational experiments in [14] indicate that this approach efficiently solves problems with up to 50 nodes. Additionally, it was found that depending on the quality of the logical topology provided as an input, the final result may differ by up to 15 % improvement in the network throughput.

An alternative heuristic approach for solving the WDM optical network design problem was used by Prathombutr et al. [20]. It simplifies the LTD problem by pre-specifying a small number of lightpaths that can be used for the LTD problem. In [20], the set of lightpaths that can be used for the LTD problem was generated either randomly, or by solving the k-shortest path problem in the physical topology. The proposed solution procedure, a multi-objective evolutionary algorithm (MOEA), performs a search for good solutions by defining chromosomes with a feasible logical topology. The routing of traffic over the established logical topology is then determined using a shortest path algorithm. Computational experiments performed on a 6-node network and the NSFNET network for a single set of traffic demands, no bifurcation of flow, and varying number of transmitters and receivers and number of wavelengths available at each fiber, indicate that MOEA outperforms two heuristics presented by Zhu and Mukherjee [31].

Banerjee and Mukherjee [5] proposed a MIP defined on a small subset of all possible lightpaths. For each (s, d) pair, only those variables that represented lightpath (s, d) over physical edges contained in the k-shortest paths were included in the formulation. Similarly, for each (s, d) pair only those variables that represented traffic carried between nodes contained in the k-shortest paths were included in the formulation. The computational experiments performed in [5] included three networks: the 14-node NSFNET network, a 15-node PACBELL network, and a 20-node randomly generated network. In all tests, it was assumed that there is only a single fiber on every physical link and that the number of transmitters is equal to the number of receivers and constant across the network.

A similar two-layer network design problem appears in ATM networks. For example, Ball and Vakhutinsky [2, 3] consider a two-layer network design problem where there are no node equipment requirements, and at least a portion of each traffic request needs to be served. For this problem, Ball and Vakhutinsky [2] proposed two models for networks that are (and are not, respectively) fault tolerant to single link failures. Given the computational difficulty of this problem, Ball and Vakhutinsky [2] proposed several improvement and simplification strategies, including variable aggregation, use of the linear relaxation instead of the complete MIP formulation, development of heuristic procedures that take advantage of the results obtained by solving the linear relaxation, and development of valid inequalities that strengthen the linear relaxation. The computational tests on two network configurations, a grid network with 8 nodes and 10 undirected links, and a geographically distributed network with 12 nodes and 20 undirected links, indicated that the proposed procedure provides results that have an integrality gap between 1.86 and 10.53 %.

Dahl et al. [9] considered a general problem for the layered telecommunication networks. They studied a two-layer network design problem, where the network is defined by a physical network $N = (V, L)$, and a pipe graph $G = (V, E)$ that consists of a pre-specified set of pipes installed over the edges of graph N. In this problem, the objective is to route all the traffic, so that each commodity uses exactly one pipe path, and the fixed pipe installation cost and the cost of routing flows over pipes is minimized. Additional restrictions include a limit on the number of pipes that may be established over any given physical edge, and, a limit on the capacity of pipes used. So, the sum of all flows using a given pipe must be less than its capacity. They also assumed that the demand of any given commodity could only have one of two possible values. The proposed solution procedure is an efficient branch-and-cut algorithm that uses knapsack and hypomatchable inequalities. The computational experiments performed on problems with up to 62 nodes and 81 edges in the physical graph indicated that when the pipe installation cost is set to very small values, the proposed branch-and-cut algorithm performs quite well (all the test problems were solved to optimality at the root node in quite short CPU times). However, the procedure turned out to be less effective in the case of high pipe installation cost.

Belotti and Malucelli [7] proposed a column generation algorithm for the two-layer telecommunications network design, where the traffic can be routed over a combination of individual physical edges and virtual paths (referred to as semi-paths in [7]). The only constraints considered in this problem are the capacities of the semi-paths and the individual physical edges (if used instead of semi-paths). The column generation procedure starts with a subset of lower and higher layer paths and progressively solves the problem. If the solution found by the column generation procedure is fractional, a heuristic rounding algorithm is applied in search for a feasible integer solution. The column generation procedure is then repeated using the modified reduced costs that favor the use of physical edges that are not close to their capacity.

Current literature suggests that there has not been much work in terms of exact procedures for the "simultaneous" logical topology design and traffic routing. One such procedure was developed by Sung and Song [27], where they proposed the first branch-and-price algorithm that simultaneously considers traffic routing and logical topology design. However, the optical network design problem studied in [27] does not impose any restrictions at the network nodes in terms of equipment available, and the capacity of lightpaths is not considered to be a constraint. It is also assumed that at most one logical path can be established between any two nodes in the network. Sung and Song use these assumptions to develop very simple branching procedure that has no impact on the subproblem solved in column generation. However, the branch-and-price procedure developed in [27] is based on an incorrect assertion that the unit integer multicommodity flow problem is polynomially solvable. This is not true in general, and it can be shown that this problem is NP-hard (see [13]) even on tree networks.

In this paper, we propose an exact branch-and-price procedure for the design of more general WDM optical networks, where the number of lightpaths that can be established between any two nodes in the network—and their routing over the physical network—is not pre-specified. We show that this type of network setting requires careful choice of branching decisions, and propose a branching strategy that works well with the corresponding column generation algorithm. Additionally, we take into account typical node requirements in WDM optical networks in terms of the number of transmitters and receivers available at each node. We also include a realistic assumption that lightpaths that can be established in the network have a limited capacity. Our focus is on all-optical WDM networks which is the direction in which the technology is moving. Hence the wavelength continuity constraint is relaxed with the assumption that optical wavelength convertors are available. As such our model assumptions are quite general and represent a two-level multicommodity flow problem (where the flow on the higher level can be bifurcated).

3 Mathematical Formulation of the WDM Optical Network Design Problem

One way to formulate the WDM optical network design problem considered in this paper is to use arc-based variables for the definition of the logical topology. These variables simply tell us the number of lightpaths between a pair of nodes, but do not tell us their routes over the physical network. We also need to define a set of multicommodity flow variables that determine how traffic is routed over this logical topology (i.e., over the network defined by these arc-based variables that define the logical topology). Finally, we need to determine how the lightpaths are routed over the physical network. For this, we need to define a set of multicommodity flow variables that determine how the logical topology variables (lightpaths) are routed over the physical network. The problem with this arc-based formulation is that it has 3 sets of arc-based variables and rapidly becomes computationally intractable as the size of the network increases. Consequently, we develop a column generation approach for the WDM optical network design problem. This approach works on the two layers of WDM optical networks, using the path-based mathematical formulation that we describe next.

In the following discussion, we assume that individual traffic requests do not exceed the total payload (i.e. capacity) of a lightpath. All traffic requests are, accordingly, scaled by the capacity of a single lightpath, so that a bandwidth requirement of one traffic unit (TU) is equal to the capacity of a lightpath. In our model, we differentiate between a lightpath variable where we only know the origin and destination of the lightpath and not it's routing through the physical network, and a lightpath variable where we know it's routing through the physical network. It is convenient to do so because routing of the traffic over the logical topology does not require knowledge of how the lightpaths are routed over the physical network. Thus, as we shall see, it allows for independent pricing of variables in our column generation procedure. We refer to lightpath variables where we only know the origin and destination and not it's routing over the physical network as **virtual lightpath variables**, and refer to lightpath variables where we know it's routing over the physical network as **real lightpath variables**.

The notation that we use is as follows. The set $G = (V, A)$ represents the physical topology defined over a set of nodes V connected by a set of arcs (direct fiber connections) A. The set $T^{(s,d)}$ represents the total (scaled) demand from node s to d, and the set Ω represents the set of all demands in the network. The set Λ represents the set of all possible lightpath origin-destination pairs. These correspond to all possible virtual lightpaths. The set Z represents the set of all possible real lightpaths in the network (all the lightpaths in set Z have fully specified paths in the physical topology). The set $P^{(s,d)}$ represents the set of all possible flow paths p for any given commodity (s, d). The number of wavelengths available between nodes i

and j that are directly connected by optical fibers is denoted by L_{ij} (when multiple fibers are available, this number represents a product of the number of fibers and the number of wavelengths available at each fiber). The number of transmitters available at any given node i is denoted by Δ_t^i, and the number of receivers available at any given node j is denoted by Δ_r^j.

We also use the following four groups of variables. The virtual lightpath variables, $Y^{(i,j)}$, indicate the number of lightpaths established between nodes i and j (note that these variables do not provide information on how the lightpaths between a given origin and destination are routed over the physical topology). The number of lightpaths of type $z \in Z$ used in a given solution is specified by the real lightpath variables, $X_z^{(i,j)}$ (these variables tell us which specific lightpaths from the set Z are used in the final solution). The flow path variables, $f_p^{(s,d)}$ indicate the fraction of the traffic demand for the commodity (s, d) carried on flow path p. In this formulation, flow paths are defined over the virtual lightpath variables (i.e., the flow path variable tells us how traffic between an origin s and destination d is routed in the logical topology defined by the lightpaths). Finally, the lost traffic variables, $H^{(s,d)}$, indicate the fraction of demand of commodity (s, d) that is lost or not satisfied.

MIP-PATH Formulation

$$\text{Minimize} \quad \sum_{(s,d)\in\Omega} T^{(s,d)} H^{(s,d)} \tag{1}$$

$$\text{subject to} \quad \sum_{j:(i,j)\in\Lambda} Y^{(i,j)} \le \Delta_t^i \quad \forall i \in V \tag{2}$$

$$\sum_{i:(i,j)\in\Lambda} Y^{(i,j)} \le \Delta_r^j \quad \forall j \in V \tag{3}$$

$$Y^{(i,j)} - \sum_{z:O(z)=i,D(z)=j} X_z^{(i,j)} = 0 \quad \forall (i,j) \in \Lambda \tag{4}$$

$$\sum_{(i,j)\in\Lambda, z:(l,m)\in z} X_z^{(i,j)} \le L_{lm} \ \forall (l,m) \in A \tag{5}$$

$$Y^{(i,j)} - \sum_{p:(i,j)\in p} f_p^{(s,d)} \ge 0 \quad \forall (i,j) \in \Lambda, (s,d) \in \Omega \tag{6}$$

$$Y^{(i,j)} - \sum_{(s,d)\in\Omega, p:(i,j)\in p} T^{(s,d)} f_p^{(s,d)} \ge 0 \quad \forall (i,j) \in \Lambda \tag{7}$$

$$\sum_{p\in P^{(s,d)}} f_p^{(s,d)} + H^{(s,d)} = 1 \quad \forall (s,d) \in \Omega \tag{8}$$

$$f_p^{(s,d)} \in R_+^1 \ \forall p \in P^{(s,d)}, (s,d) \in \Omega \tag{9}$$

$$Y^{(i,j)} \in R_+^1 \ \forall (i,j) \in \Lambda \tag{10}$$

$$X_z^{(i,j)} \in Z_+^1 \ \forall z \in Z, (i,j) \in \Lambda \tag{11}$$

Constraints (2) and (3) in this formulation limit the out/in degree of any node to be not larger than the total number of transmitters/receivers. Constraint set (4) ensures that all virtual lightpaths are defined in the physical topology through an adequate number of real lightpaths in the physical topology. Since the real lightpath variables are defined as integers, these constraints also guarantee that the $Y^{(i,j)}$ variables are integer as well, and can therefore be defined as real variables (constraint (10)). Constraint set (5) represents a limit on the number of real lightpaths that can be established on any physical edge. Constraint set (6) ensures that the flow path of any given commodity (s, d) can use virtual lightpath (i, j) only if that virtual lightpath is included in the logical topology. Constraints (7) are capacity constraints limiting total flow over all virtual lightpaths established between two nodes. Constraint set (8) indicates that the fraction of demand for a given commodity is satisfied plus the fraction that is lost equals one. Note that constraint (6) is redundant in the presence of constraint (7), however use of this constraint significantly strengthens the linear programming (LP) relaxation of the formulation.

We note that it is quite challenging to further strengthen the linear programming relaxation of this formulation. The reason is two-fold. First, the logical topology design problem defined by constraints (2) through (5) is a more general version of the unit integer multicommodity flow problem which is not suitable for standard types of integer multicommodity flow strengthening cuts such as lifted cover inequalities. Second, we do not know which commodities will be served in the end. If we did, we could, for example, derive a simple class of valid inequalities that define the minimum number of lightpaths that need to originate and terminate at any given node in the network. In other words, if all the traffic originating from a given node i and terminating at node j must be served, then we can add the following type of valid inequalities for the nodes i and j.

$$\sum_{i:(i,j)\in\Lambda} Y^{(i,j)} \geq \left\lceil \sum_{d:(i,d)\in\Omega} T^{(i,d)} \right\rceil \qquad \forall i \in V : \exists (s,d) \in \Omega : s = i$$

$$\sum_{j:(i,j)\in\Lambda} Y^{(i,j)} \geq \left\lceil \sum_{j:(s,j)\in\Omega} T^{(s,j)} \right\rceil \qquad \forall i \in V : \exists (s,d) \in \Omega : d = j.$$

One issue related to the MIP-PATH formulation is that this formulation cannot be easily modified for use in the networks where bifurcation of flow is not allowed. To ensure no bifurcation of flow, we need to define flow variables as binary variables **and** either add "packing" constraints that would guarantee that there is no bifurcation of the flow among the lightpaths with the same origin and destination, or redefine flow variables so that they are exactly mapped to the real lightpath variables used. It turns out that this issue significantly changes the formulation and further complicates development of the branch-and-price techniques for WDM optical network design. We discuss the WDM optical network design problem without flow bifurcation and propose alternative branch-and-price procedures in a companion paper [21].

4 Branch-and-Price Algorithm for WDM Optical Network Design

Branch-and-price algorithms are procedures commonly used to solve problems with a large number of variables. They combine the linear programming concept of column generation and integer programming concept of branch-and-bound. Column generation itself is based on a very simple but powerful idea. It involves the iterative solution of a master problem that includes only a subset of the original variables, and a subproblem which represents a check on whether any other variables need to be added to this model in order to find the optimal solution to the original problem.

The effectiveness of column generation when combined with branch-and-bound to solve integer programs, depends on many implementation issues. Some of these issues include initialization, strength of the linear relaxation of the master problem, subproblem structure, branching strategies, and termination (see Vanderbeck [29], Vanderbeck and Wolsey [30], or the collection on column generation [10] for a more comprehensive review of other important issues).

The speed of a branch-and-price algorithm is often directly related to the strength of the linear relaxation of the master problem. So, it is very important to start with a strong linear relaxation of the master model. The subproblem structure has considerable importance in column generation, since, in some cases, the subproblem may turn out to be NP-hard problem itself, which then additionally complicates the column generation procedure. Ideally, we should try to formulate the master problem in such a way that the calculation of the reduced cost or solving the subproblem is (if possible) relatively easy. In the case of branch-and-price algorithms, we also need to try and ensure that the branching decisions are such that the structure of the subproblem is not significantly destroyed in the subsequent iterations.

4.1 Branch-and-Price Algorithm for the MIP-PATH formulation

One of the advantages of the MIP-PATH formulation is that all virtual lightpath variables can be included in the master model at the beginning of the search, and as flow path variables ($f_p^{(s,d)}$) are related only to the virtual lightpath variables they can be priced out separately and independently from the real lightpath variables ($X_z^{(i,j)}$). Consequently, the pricing part of the column generation algorithm can be performed in a traditional manner. We describe the details of this procedure next.

The following dual variables correspond to the constraints of the MIP-PATH formulation:

- nonpositive a_i and b_j variables for constraints (2) and (3) respectively
- unrestricted in sign $g^{(i,j)}$ variables for constraint (4)
- nonpositive $d_{(l,m)}$ variables for constraint (5)
- nonnegative $r_{(i,j)}^{(s,d)}$ variables for constraint (6)
- nonnegative $v_{(i,j)}$ variables for constraint (7)
- unrestricted in sign $w^{(s,d)}$ variables for constraint (8)

The reduced cost of any $X_z^{(i,j)}$ (real lightpath) variable is:

$$RC_z^{(i,j)} = g^{(i,j)} - \sum_{(l,m)\in z} d_{(l,m)}.$$

The reduced cost of any $f_p^{(s,d)}$ variable is:

$$RC_p^{(s,d)} = \sum_{(i,j)\in p} \left(r_{(i,j)}^{(s,d)} + T^{(s,d)} v_{(i,j)} \right) - w^{(s,d)}.$$

The new real lightpath variables that need to be added to the restricted master problem are identified by finding paths for each node pair (i, j) that minimize the following expression:

$$g^{(i,j)} - \sum_{(l,m)\in z} d_{(l,m)}$$

or equivalently that minimizes

$$-\sum_{(l,m)\in z} d_{(l,m)}.$$

This can be solved as a shortest path problem on an auxiliary network where the cost of an arc (l, m) is $-d_{(l,m)}$. If the cost of this path is less than $-g^{(i,j)}$ (implying that $g^{(i,j)} - \sum_{(l,m)\in z} d_{(l,m)} < 0$) the real lightpath variable is added. Otherwise, no new real lightpaths with origin at node i and destination at node j are added.

The new flow paths that need to be added to the restricted master problem are identified by finding the path for each commodity (s, d) with the minimum value of:

$$\sum_{(i,j)\in\Lambda} \left(r^{(s,d)}_{(i,j)} + T^{(s,d)} v_{(i,j)} \right) - w^{(s,d)}$$

or equivalently with the minimum value of

$$\sum_{(i,j)\in\Lambda} \left(r^{(s,d)}_{(i,j)} + T^{(s,d)} v_{(i,j)} \right).$$

This can be solved as a shortest path problem on an auxiliary network where the cost of an arc (i, j) is defined by the expression $r^{(s,d)}_{(i,j)} + T^{(s,d)} v_{(i,j)}$. If the cost of the shortest path is less than $w^{(s,d)}$ the flow path is added. Otherwise, no new flow paths for commodity (s, d) are added. Observe that the pricing of the flow path variables and the real lightpath variables are done independently, and on separate auxiliary networks. This is one of the advantages of the MIP-PATH formulation in this paper due to the fact that we have both virtual and real lightpath variables.

4.1.1 Branching Strategy

Although only real lightpath variables $X^{(i,j)}_z$ need to be defined as integers in the MIP-PATH formulation, our branching strategy uses virtual lightpath variables as well. Branching on the virtual lightpath variables does not significantly affect the structure of the pricing problem, which is a significant advantage. The branching is performed hierarchically from the higher network layer to the lower network layer. In other words, we first branch on the virtual lightpath variables, and then on the real lightpath variables. For the virtual lightpath variables, variable dichotomy is used, while branching decisions for the real lightpath variables are somewhat more complicated.

In order to understand the problem related to branching on real lightpath variables, consider what would happen if we performed branching based on the variable dichotomy. Essentially, for a given fractional variable $X^{(i,j)}_z$, and its current fractional value $X^{(i,j)*}_z$, we would create 2 new nodes by adding the restriction $X^{(i,j)}_z \leq \lfloor X^{(i,j)*}_z \rfloor$ on one branch, and the restriction $X^{(i,j)}_z > \lceil X^{(i,j)*}_z \rceil$ on the other branch. Enforcing the second restriction is not a problem as it indicates that we must use a particular real lightpath variable. However, the first restriction requires that we do not generate any new real lightpath variables in the column generation procedure that would have the same propagation path as a lightpath z corresponding to variable $X_z^{(i,j)}$. As noted by Barnhart et al. [6], the latter type of branching restriction is hard to enforce since there is no guarantee that after solving the shortest path problem in the subsequent iterations of column generation, we would not end up with a path (or real lightpath variable) that (due to our previous branching decisions) is not supposed to be added to the model.

One way to resolve this problem is to solve the $k+1$-shortest path problem whenever, due to the previous branching decisions, the first k-shortest path are not eligible for addition to the model. In this paper, we propose a different branching strategy that indirectly forces all real lightpath variables to be integer. This strategy is based on the observation that if the **sum** of all real lightpath variables with the same origin and destination is integer over every arc traversed in the physical layer then either:

(a) all real lightpath variables are integer, or
(b) all real lightpath variables do not have integer values, but the solution can be interpreted as integer.

The first case is trivially correct, since if all real lightpath variables are integer, it directly follows that the **sum** of all real lightpath variables with the same origin and destination over all arcs in the network is integer as well. The second case, however, is not so straightforward. First note that if the sum of all real lightpath variables with the same origin and destination is integer over all arcs in the network, and there are fractional real lightpath variables, then it must mean that there is at least one pair of nodes with at least 4 fractional real lightpath variables. (If there was only one fractional real lightpath variable, the sum of all real lightpath variables with the same origin and destination would not be integral. Similarly, if there were only two fractional real lightpath variables, these real lightpath variables would either have to follow the same path in the physical layer, which is not possible, or, the sum of all real lightpath variables over all the arcs in the network would not be integral. In other words, there must be at least 2 fractional real lightpath variables using the given arc at the same time, or none. Since 2 real lightpath variables cannot have identical propagation paths in the physical layer, it follows that we must have at least 4 fractional real lightpath variables.) An example of situation where there are 4 fractional real lightpath variables is shown in Fig. 1. Lightpath L_1 uses path (1, 3, 5), lightpath L_2 uses path (1, 3, 4, 5), lightpath L_3 uses path (1, 2, 3, 4, 5), and lightpath L_4 uses path (1, 2, 3, 5). Values of the associated real lightpath variables in a given solution are 0.7 for lightpath L_1 and L_3, and 0.3 for lightpath L_2 and L_4. Since the flow balance constraints guarantee that sum of all "flows" into any given transient node must equal the sum of all "flows" going out of that node, it follows that in any fractional solution that satisfies the requirement of an integral sum of the real lightpath variables over all arcs, we can eliminate some of the fractional real lightpath variables so that an integral solution is obtained, without any impact on other aspects of the current solution. In the example shown in Fig. 1, we could, for example, eliminate lightpaths L_2 and L_4, and round up the values of the variables associated with lightpaths L_1 and L_3, without making any other changes to current solution. This observation

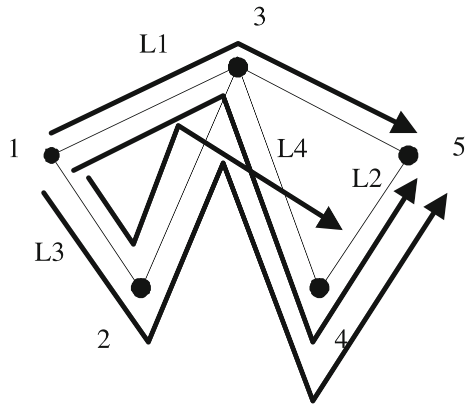

Fig. 1 Example of branching issue for the MIP-PATH formulation

allows us to branch on the **sum** of all real lightpath variables with the same origin and destination that are using same arc in the physical layer, instead of direct branching on individual real lightpath variables.

In summary, whenever the solution at a given node of the branch-and-bound tree is fractional, we perform the following steps.

Step 1. Check if there are any fractional virtual lightpath variables. If there are no such variables, go to Step 2, otherwise, select one fractional virtual lightpath variable, and create 2 new nodes using variable dichotomy.

Step 2. Check if there is an arc with a fractional sum of all real lightpath variables with the same origin and destination using that arc. If there is no such arc, exit, otherwise create 2 new nodes:

Node 1. Add restriction $\sum_{z:(m,n)\in z} X_z^{(i,j)} \leq \lfloor \sum_{z:(m,n)\in z} X_z^{(i,j)*} \rfloor$

Node 2. Add restriction $\sum_{z:(m,n)\in z} X_z^{(i,j)} \geq \lceil \sum_{z:(m,n)\in z} X_z^{(i,j)*} \rceil$

where (m, n) and (i, j) are an arc and lightpath origin-destination pair identified in the previous check, and $X_z^{(i,j)*}$ are the actual values of real lightpath variables in the current solution.

One problem with this approach is that the dual variable associated with the constraint added in Node 2 is positive, and thus may lead to the creation of negative-cost arcs which in turn could result in a negative-cost cycle. In the presence of a negative-cost cycle the shortest (loopless) path problem is NP-complete, and thus is no longer polynomially solvable. In our computational experiments, however, we did not encounter the negative cost arc problem.[2] Thus, we use a standard shortest path

[2]We note that negative cost cycles rarely appear in our pricing problem. The reason for this is threefold. First, the dual variable associated with a Node 2 constraint is only positive when constraint (5) associated with the same arc is near its capacity. Second, the cost of arcs in the corresponding graph is comprised of both non-negative and non-positive dual variables, meaning that the arc cost will not necessarily have a negative value even if the dual variable associated with a Node 2 constraint is positive. Third, the existence of a negative-cost arc by itself does not necessarily imply the existence of a negative cost cycle.

algorithm to solve the pricing problem. When a negative cost cycle is encountered, we resort to solving the pricing problem via integer programming. (The shortest loopless path problem between a source (s) and a destination (d) may be formulated as a minimum cost network flow problem where we wish to send one unit of flow from s to d with the additional restrictions that the flow into any intermediate node is at most 1, and all flow is integer.)

4.2 Applying the Branch-and-Price Algorithm for MIP-PATH to WDM Optical Network Design with Alternative Design Objectives

In situations where we do not need information on propagation of the flow paths in the physical layer, we can easily apply our branch-and-price algorithm for MIP-PATH. For example, in the case where we need to minimize the number of transponders in the network, the objective function is:

$$Min \quad \sum_{i} \left(\Delta_t^i + \Delta_r^i\right)$$

where Δ_t^i and Δ_r^i represent variables corresponding to the number of transmitters and receivers used at each node in the network. The use of this objective function would only mean that Δ_t^i and Δ_r^i would be variables, not given constants. This has **no** effect on our branch-and-price algorithm, since integrality of $Y^{(i,j)}$ variables immediately implies that Δ_t^i and Δ_r^i will be integer as well, i.e., there is no need to branch on these variables in order to get integer values.

5 Upper Bounding Heuristics for WDM Optical Network Design

The following two heuristic procedures for the WDM optical network design problem represent extensions of the heuristic logical topology design algorithm (HLDA) proposed by Ramaswami and Sivarajan [22]. The initial steps of the first heuristic procedure are identical to that of the original HLDA. In other words, we first try to establish direct lightpaths that will serve the commodities with the largest traffic (in other words, we try to establish a single lightpath for each commodity in the network). If we cannot serve all the commodities with the direct lightpaths, we attempt to establish additional lightpaths using any remaining capacity in the network. These additional lightpaths are randomly established in the original HLDA. We modify this step by trying to establish lightpaths that utilize as few physical hops (edges) as possible. Finally, we deal with routing of traffic that could not be served by the direct lightpaths. This final step differs from the original HLDA, which stops once the logical topology is identified. In [22], the traffic routing is determined using a linear program (LP), which solves the WDM optical network design problem over the fixed logical topology determined by HLDA. We use the same approach to try to find a better traffic routing in the network by fixing the logical topology to the one determined by our heuristic ModHLDA. We refer to this second procedure as LPModHLDA. The steps of the two proposed heuristic procedures are described next.

5.1 ModHLDA

ModHLDA works in three phases. First, commodities with the highest demand are served by direct lightpaths. Second, if there are any unused transmitters and receivers left from the first phase, additional lightpaths are established. Finally, in the third phase, an attempt is made to route all commodities that were not initially served with the direct lightpaths.

Routing Commodities over Direct Lightpaths

Step 1 Sort all commodities in decreasing order of their demand.

Step 2 Establish direct lightpaths for commodities with the highest demand first. If there is insufficient capacity in the network to serve all the commodities with the direct lightpaths, go to Step 3. Otherwise, STOP.

Establishing Additional Lightpaths

Step 3 If there are any unused transmitters and receivers left, go to Step 4. Otherwise, go to Step 7.

Step 4 Find the shortest paths (in terms of the number of physical hops) for all pairs of nodes that have unused transmitters and receivers. If candidate lightpaths are found, go to Step 5. Otherwise, go to Step 7.

Step 5 Sort the shortest paths from Step 4 in increasing order of their length.

Step 6 Establish new lightpaths, starting with the ones with the shortest length. When a lightpath cannot be established due to an insufficient number of available wavelengths go to Step 4. Otherwise go to Step 7.

Routing Unserved Commodities

Step 7 For each unserved commodity (starting from the commodities with the highest demand), find the shortest path (in terms of the number of lightpaths used) and route the commodity on it. (Note: All lightpaths with positive remaining capacity are considered in finding the shortest path of a given commodity). If the path found has capacity greater than or equal to the demand of a given commodity, all of the demand of the commodity is routed on this path and the remaining capacity on the arcs is updated. Otherwise, only a portion of the unserved demand equal to the capacity of the path is routed. In this case, after updating the remaining capacity on the arcs the procedure is repeated until all of the demand is routed, or it turns out it is not possible to route all the demand (i.e., no paths can be found for a given commodity).

5.2 LPModHLDA

LPModHLDA uses the logical topology determined in the first two phases of ModHLDA, and then solves the following linear program.

$$Min \sum_{(s,d)\in\Omega} T^{(s,d)} H^{(s,d)} \tag{12}$$

Subject to:

$$\sum_{i:(i,k,l)\in\Lambda} f_{(i,k,l)}^{(s,d)} - \sum_{j:(k,j,l)\in\Lambda} f_{(k,j,l)}^{(s,d)} = \begin{cases} 1 - H^{(s,d)} & \text{if } k = d \\ H^{(s,d)} - 1 & \text{if } k = s \quad \forall (s,d) \in \Omega, k \in V \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

$$\sum_{(s,d)\in\Omega} T^{(s,d)} f_{(i,j,l)}^{(s,d)} \leq 1 \quad \forall (i,j,l) \in \Lambda \tag{14}$$

$$f_{(i,j,l)}^{(s,d)} \in R_+^1 \ \forall (i,j,l) \in \Lambda, (s,d) \in \Omega \tag{15}$$

$$H^{(s,d)} \in R_+^1 \ \forall (s,d) \in \Omega \tag{16}$$

The variables in the above linear program include flow variables $f_{(i,j,l)}^{(s,d)}$, each indicating the amount of flow of the commodity(s, d) carried over the lth lightpath between nodes i and j, and the lost traffic variables $H^{(s,d)}$, indicating the amount of lost traffic for the commodity (s, d). The sets V, Λ, and Ω, represent (respectively) the set of nodes in the physical layer, set of lightpaths determined by the HLDA, and set of commodities. Note that, since the logical topology is fixed, we need only the flow balance constraints (13), and constraints on the payload of the lightpaths (14).

6 Computational Experiments

We coded the branch-and-price algorithm for the MIP-PATH formulation in Microsoft Visual C++, ILOG *Maestro* [16] library, and CPLEX 9.0. All computations were performed on a workstation with a 2.66 GHz Xeon processor and 2GB RAM.

We used four different sets of problems in our computational tests. The first two sets represent randomly defined problems, while the other two represent the NSFNET network and a set of problems defined and used in the study performed by Prathombutr et al. [20].

The first set of random instances is defined over a complete graph representing the physical layer, while the second set is defined over an incomplete graph representing the physical layer. For each set we generated networks with 5, 7, 10, and 20 nodes with four different traffic patterns. Two traffic patterns are defined so that there is demand between all pairs of nodes, and the other two are defined so that there is demand only between 50 % of the nodes in the network. For each "density" of traffic demand matrix we defined two demand levels - high and low. High demand indicates that the demand between pairs of nodes was uniformly generated in the interval [0.1, 1], while low demand indicates that the demand between pairs of nodes was uniformly generated in the interval [0.1, 0.5]. The second set of instances is generated in the same manner as the first set, only, in this case we also randomly generated the physical network. (The physical network in this case is generated using a uniform $[1, |V|]$ random distribution to select the pairs of nodes that will be connected by a pre-specified number of arcs. Similarly, we have used a uniform distribution to determine commodities with a positive demand.)

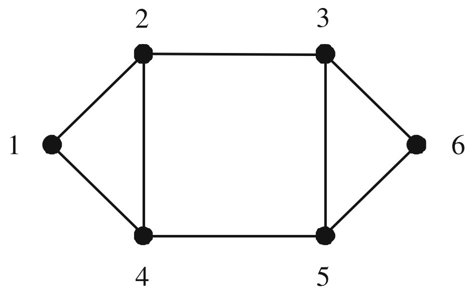

Fig. 2 Physical topology of the 6-node optical network

The third and the fourth set of test instances are from the study of Parthombutr et al. [20]. These include a simple 6-node network and the NSFNET network. Since instances defined in [20] use "granular" traffic in terms of the number of OC-1, OC-3, and OC-12 demands between pairs of nodes, we have modified these instances so that the total demand between two nodes is expressed as a fraction of the capacity of a lightpath, which was set to OC-48 in the 6-node network, and OC-192 in the NSFNET network. In one instance (commodity (5, 1)), we have reduced original demand from 54 OC units to 48 OC units to accommodate for our assumption that demand between pairs of nodes does not exceed the payload of a lightpath. As in [20], 7 instances of 6 nodes network, and 6 instances of NSFNET network are tested for different number of wavelengths that can be supported by a single fiber and a different maximum number of transmitters and receivers that can be used at each node in the network. The physical topology of both networks is shown in Figs. 2 and 3 respectively, and the corresponding modified demand matrices are shown in Tables 1 and 2. For these instances, we performed tests using the design objective that minimizes the total lost traffic in the network.

In all our test instances, the column generation is started with a single lightpath defined for each pair of nodes (determined using the shortest path algorithm, with the length of the path defined by the number of physical hops used by the lightpath), and a single flow path for each commodity. Additionally, our initial computational

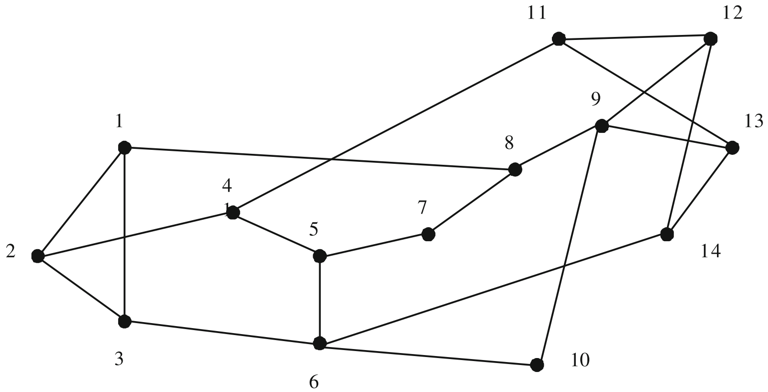

Fig. 3 Physical topology of the NSFNET optical network

Table 1 Demand matrix for 6-node network example in Fig. 2

	1	2	3	4	5	6
1	0.00	0.73	0.46	0.54	0.56	0.44
2	0.75	0.00	0.92	0.73	0.77	1.13
3	0.35	0.69	0.00	0.69	0.50	0.77
4	0.90	0.67	1.00	0.00	0.65	0.48
5	0.83	0.79	0.38	0.56	0.00	0.56
6	0.60	0.52	0.92	0.94	0.77	0.00

experiments indicated that the use of the minimum parent lower bound in the node selection phase of the branching provides better results than the depth first rule (in other words, the node on which we perform column generation and branching next is the node that has a minimum parent lower bound in the branch-and-bound tree at that point). We have, therefore, used the minimum parent lower bound for selection of branching nodes in all of our computational experiments.

In order to improve the upper bounds obtained by our branch-and-price algorithms, we have included the use of the local (default CPLEX) MIP optimizer at the root node of the branch-and-bound tree with a 600 sec CPU time limit.

Table 3 provides the results for the randomly defined instances that have a complete graph in the physical layer. We can see that LPModHLDA provides significantly better upper bounds than ModHLDA with only a small increase in computation time. Specifically, the upper bounds provided by the LPModHLDA procedure are 3.54 % better than those provided by the ModHLDA procedure, while the average CPU times for the ModHLDA and LPModHLDA procedures were 0.02 and 1.26 sec respectively. Our branch-and-price algorithm, on the other hand, required 161.53 sec on average to find the optimal solutions for 11 out of 16 instances. The gap between the optimal solutions found by our branch-and-price algorithm and the LPModHLDA procedure ranged between 0 and 7.80 %, and, on average, was 1.89 %. For the 5 instances where our branch-and-price algorithm did not find the optimal solution, the average percentage gap between the lower

Table 2 Demand matrix for the NSFNET network shown in Fig. 3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0.00	0.11	0.03	0.15	0.08	0.17	0.14	0.13	0.18	0.09	0.13	0.19	0.26	0.17
2	0.22	0.00	0.20	0.26	0.06	0.18	0.09	0.22	0.09	0.14	0.10	0.04	0.14	0.10
3	0.15	0.13	0.00	0.21	0.18	0.13	0.02	0.29	0.16	0.17	0.27	0.17	0.06	0.18
4	0.05	0.08	0.13	0.00	0.28	0.13	0.11	0.24	0.16	0.22	0.14	0.27	0.20	0.17
5	0.28	0.06	0.26	0.23	0.00	0.12	0.21	0.27	0.22	0.17	0.16	0.21	0.14	0.01
6	0.29	0.14	0.32	0.07	0.25	0.00	0.20	0.15	0.23	0.11	0.06	0.16	0.08	0.28
7	0.18	0.09	0.14	0.13	0.18	0.24	0.00	0.17	0.17	0.07	0.16	0.18	0.19	0.19
8	0.10	0.12	0.07	0.19	0.11	0.32	0.20	0.00	0.13	0.08	0.22	0.29	0.04	0.10
9	0.24	0.15	0.23	0.26	0.11	0.15	0.28	0.08	0.00	0.26	0.11	0.13	0.16	0.26
10	0.22	0.07	0.06	0.17	0.17	0.15	0.17	0.06	0.03	0.00	0.14	0.28	0.27	0.27
11	0.23	0.16	0.27	0.24	0.06	0.13	0.24	0.11	0.15	0.14	0.00	0.20	0.21	0.25
12	0.33	0.13	0.21	0.19	0.16	0.17	0.07	0.21	0.14	0.13	0.06	0.00	0.05	0.15
13	0.26	0.20	0.21	0.21	0.16	0.15	0.03	0.17	0.17	0.16	0.21	0.11	0.00	0.24
14	0.25	0.20	0.11	0.09	0.19	0.16	0.08	0.09	0.18	0.19	0.15	0.08	0.27	0.00

Table 3 Minimizing the lost traffic in the network

$\|V\|$	$\|C\|$	D	B&P Algorithm			ModHLDA		LP ModHLDA	
			LB	UB	CPU (sec)	UB	CPU (sec)	UB	CPU (sec)
5	20	H	0.11	0.11	0.39	0.11	0.02	0.11	0.02
5	20	L	0.00	0.00	0.09	0.00	0.02	0.00	0.02
5	10	H	0.00	0.00	0.06	0.00	0.02	0.00	0.02
5	10	L	0.00	0.00	0.06	0.00	0.00	0.00	0.02
7	42	H	3.95	3.95	17.44	5.12	0.02	4.40	0.03
7	42	L	0.00	0.00	1.25	0.00	0.02	0.00	0.03
7	21	H	0.19	0.19	46.06	1.12	0.02	0.99	0.02
7	21	L	0.00	0.00	0.31	0.00	0.02	0.00	0.02
10	90	H	21.15	21.15	390.11	23.28	0.02	22.66	0.11
10	90	L	5.20	5.20	1104.19	8.48	0.02	6.19	0.14
10	45	H	2.54	2.79	3603.38	3.85	0.02	3.35	0.06
10	45	L	0.00	0.00	216.81	0.00	0.02	0.00	0.05
20	380	H	152.44	152.45	3653.25	154.66	0.03	153.87	8.48
20	380	L	70.49	70.49	3657.91	76.27	0.03	70.72	7.08
20	190	H	52.76	53.43	3623.09	56.66	0.03	55.76	2.13
20	190	L	16.60	18.99	3671.94	24.68	0.03	19.97	1.89

Complete physical network with 2 fibers (fiber capacity: 2 lightpaths) between all pairs of nodes

and upper bound at the end of the search was 1.78 %.[3] In these five instances the average improvement of the upper bound of our branch-and-price algorithm over the LPModHLDA procedure was 2.6 %.

Table 4 provides the results for the randomly defined instances that have an incomplete graph in the physical layer. In this case, the LPModHLDA procedure provided upper bounds that were 0.62 % better than those provided by the ModHLDA procedure. However, the upper bounds found by the LPModHLDA procedure were on average 0.49 % from the optimal solutions found in 5 instances by our branch-and-price algorithm. In the three instances where our branch-and-price algorithm did not find the optimal solution, the average gap between the lower and upper bound was 3.46 %. In these instances, the average improvement in the upper bound over the solutions provided by the LPModHLDA procedure was 0.11 %.

The results for test instances defined in the study of Prathombutr et al. [20] are shown in Tables 5 and 6. We can see that within a specified CPU time limit, our branch-and-price algorithm found optimal solutions in all instances of the 6 node network, and in one instance of the NSFNET network. In the 6 node network instances, the average optimality gap for the LPModHLDA procedure was 2.99 %, while the average improvement of the LPModHLDA procedure over the ModHLDA procedure upper bounds was 1.3 %. In the case of the NSFNET instances, our branch-and-price algorithm did not provide good upper bounds, and the average optimality gap was 11.17 %. This can be explained by the long time

[3]The percentage gaps in these tests were calculated as $\frac{UB-LB}{total\ demand-UB} * 100\,\%$. In other words, we provide the percentage gaps with the respect to the total traffic served, instead of the total traffic lost. (We do this only because the straightforward computation of the gap in the form $\frac{UB-LB}{LB}$ is not practical for these problems, since the LB is equal to zero in a large number of instances.)

Table 4 Minimizing the lost traffic in the network

$\|V\|$	$\|A\|$	$\|C\|$	D	B&P Algorithm			ModHLDA		LP ModHLDA	
				LB	UB	CPU (sec)	UB	CPU (sec)	UB	CPU (sec)
5	10	15	H	0.38	0.38	0.20	0.38	0.00	0.38	0.02
5	10	15	L	0.00	0.00	0.09	0.00	0.02	0.00	0.02
7	15	30	H	2.04	2.04	379.58	2.66	0.02	2.40	0.03
7	15	30	L	0.00	0.00	0.69	0.00	0.02	0.00	0.03
10	20	40	H	1.85	2.11	3603.26	2.81	0.02	2.69	0.06
10	20	40	L	0.00	0.00	315.98	0.00	0.02	0.00	0.03
20	30	60	H	4.68	5.79	3609.28	5.83	0.05	5.68	0.22
20	30	60	L	0.00	0.92	3636.55	0.91	0.05	0.56	0.09

Incomplete physical network with 2 fibers (fiber capacity: 20 lightpaths) on all arcs

Table 5 Minimizing the lost traffic in the network

T/R Nbr	Wav. Nbr	B&P Algorithm			ModHLDA		LP ModHLDA	
		LB	UB	CPU (sec)	UB	CPU (sec)	UB	CPU (sec)
3	3	4.67	4.67	5.98	5.20	0.02	4.76	0.03
4	3	0.63	0.63	5.56	1.88	0.02	1.35	0.03
5	3	0.28	0.28	13.52	1.17	0.02	1.17	0.03
7	3	0.28	0.28	4.89	0.28	0.02	0.28	0.05
3	4	4.67	4.67	6.22	6.11	0.02	5.70	0.03
4	4	0.63	0.63	25.38	1.87	0.02	1.78	0.03
5	4	0.00	0.00	43.36	0.00	0.02	0.00	0.03

The 6-node network with a single fiber on each arc

Table 6 Minimizing the lost traffic in the network

T/R Nbr	Wav. Nbr	B&P Algorithm			ModHLDA		LP ModHLDA	
		LB	UB	CPU (sec)	UB	CPU (sec)	UB	CPU (sec)
3	3	3.99	6.92	3616.56	9.50	0.03	6.27	0.86
4	3	0.00	5.35	3620.70	2.99	0.03	0.35	1.02
5	3	0.00	1.85	3624.09	0.36	0.03	0.00	0.67
4	4	0.00	4.98	3619.94	2.91	0.03	0.88	1.20
5	4	0.00	1.68	3613.88	0.00	0.05	0.00	0.97
6	4	0.00	0.00	2103.63	0.00	0.03	0.00	1.00

The NSFNET network with a single fiber on each arc

Table 7 Summary: minimizing the lost traffic in the network

Net. type	Finite wavelengths		Infinite wavelengths	
	Avg. gap (%)	Avg. CPU (sec)	Avg. gap (%)	Avg. CPU (sec)
Complete	0.56	1249.15	0.43	1222.53
Incomplete	1.30	1443.20	1.05	1421.12
6 node	0.00	14.99	0.00	7.87
NSFNET	11.17	3366.47	5.86	2938.91
Overall	2.33	1400.96	1.36	1314.00

Branch-and-price results for WDM networks with and without constraint on the number of wavelengths in the network needed to solve the LP at each node of the branch-and-bound tree. In these instances, the LPModHLDA procedure provided solutions that were on average 5.74 % better than the ones provided by the ModHLDA procedure. Additionally, the solutions provided by the LPModHLDA procedure were better than the upper bounds found by our branch-and-price algorithm. The average improvement provided by the LPModHLDA procedure in this case was 8.7 %.

We also tested the performance of our branch-and-price algorithm for the special case of WDM optical networks with an infinite number of wavelengths. This is a special case of the WDM optical network design problem that arises in situations when it is reasonable to assume that the number of lightpaths on a physical arc of the network does not represent an actual restriction and can be therefore relaxed.

Table 7 provides the summary of our computational tests for WDM optical networks with an infinite number of wavelengths. As expected, these results indicate a significant improvement in the performance of our branch-and-price algorithm when the constraint on the number of (real) lightpath variables an on an arc is relaxed.

7 Conclusion

The integrated design of both the logical topology of a WDM network and the traffic routing on it is an extremely challenging problem, and to our knowledge, for ease of computation, such problems have typically been broken up into two parts. In this paper we presented a branch-and-price algorithm for the integrated design of the logical topology of a WDM optical network and the traffic routing over it where bifurcation of traffic is permitted. We also proposed two simple upper bounding heuristics: the ModHLDA and LPModHLDA algorithms that can be used as fast upper bounding procedures for this problem and significantly improves upon the HLDA procedure used previously.

The results of our computational experiments performed over a set of four different optical networks with 5–20 nodes indicate that our branch-and-price algorithm solves the integrated WDM optical network design problem with an average optimality gap of 2.33 % within the pre-specified 3,600 sec CPU time limit. We have also found that in most instances our branch-and-price algorithm provides better upper bounds than the two proposed heuristic procedures, with the exception of the NSFNET optical network, where one of the proposed upper bound heuristics provided the best upper bounds. These results indicate that there are situations where our branch-and-price algorithm may benefit from the use of integrated upper bound heuristics. We leave this for future work.

References

1. Azodolmolky, S., Klinkowski, M., Marin, E., Careglio, D., Pareta, J., Tomkos, I.: A survey of physical layer impairments aware routing and wavelength assignment algorithms in optical networks. Comput. Netw. **53**, 926–944 (2009)
2. Ball, M.O., Vakhutinsky, A.: Fault tolerant virtual path layout: optimization models. In: Sanso, B., Soriano, P. (eds.) Telecommunications Network Planning, pp. 210–217. Kluwer Academic, Boston (1998)
3. Ball, M.O., Vakhutinsky, A.: Fault-tolerant virtual path layout in ATM networks. INFORMS J. Comput. **13**(1), 76–94 (2001)

4. Bandyopadhyay, S.: Dissemination of Information in Optical Networks: from Technology to Algorithms. Springer (2008)
5. Banerjee, D., Mukherjee, B.: Wavelength-routed optical networks: linear formulation, resource budgeting tradeoffs, and a reconfiguration study. IEEE/ACM Trans. Netw. **8**(5), 598–607 (2000)
6. Barnhart, C., Hane, C.A., Vance, P.H.: Using branch-and-price-and-cut to solve origin-destination integer multicommodity flow problems. Oper. Res. **48**(2), 318–326 (2000)
7. Belotti, P., Malucelli, F.: Multilayer network design: a row-column generation algorithm. In: Proceedings of International Network Optimization Conference, pp. B2.422–B2.427 (2005)
8. Cinkler, T., Coudert, D., Flammini, M., Monaco, G., Moscardelli, L., Munoz, X., Sau, I., Shalom, M., Zaks, S.: Traffic grooming: combinatorial results and practical resolutions. In: Koster, A., Munoz, X. (eds.) Graphs and Algorithms in Communication Networks, pp. 63–94. Springer (2010)
9. Dahl, G., Martin, A., Stoer, M.: Routing through virtual paths in layered telecommunication networks. Oper. Res. **47**(5), 693–702 (1999)
10. Desaulniers, G., Desrosiers, J., Solomon, M.M. (eds.): Column Generation. Springer (2005)
11. Durhuus, T., Mikkelsen, B., Joergensen, C., Danielsen, S., Stubkjaer, K.: All-optical wavelength conversion by semiconductor optical amplifiers. J. Lightwave Technol. **14**(6), 942–954 (1996)
12. Elmirghani, J., Mouftah, H.: All-optical wavelength conversion technologies and applications in DWDM networks. IEEE Commun. Mag. **38**(3), 86–92 (2000)
13. Garg, N., Vazirani, V.V., Yannakakis, M.: Primal-dual approximation algorithms for integral flow and multicut in trees. Algorithmica **18**, 3–20 (1997)
14. Haque, A., Aneja, Y.P., Bandyopadhyay, S., Jaekel, A., Sengupta, A.: Some studies on the logical topology design of large multi-hop optical networks. Opt. Netw. Mag. **3**(4), 96–105 (2002)
15. Hu, J.Q., Leida, B.: Traffic grooming, routing, and wavelength assignment in optical WDM mesh networks. In: Proceedings of the 23rd Annual Joint Conference IEEE Computer Communications Society, vol. 1, pp. 495–501. IEEE, Piscataway (2004)
16. ILOG Branch & Price & Cut Shortest Path Optimizers Prototype Manual. ILOG, France (2003)
17. Konda, V.R., Chow, T.Y.: Algorithm for traffic grooming in optical networks to minimize the number of transceivers. In: Proc. IEEE 2001 Workshop on High Performance Switching and Routing, pp. 218–221 (2001)
18. Mukherjee, B.: Optical WDM Networks. Springer (2006)
19. Mukherjee, B., Banerjee, D., Ramamurthy, S., Mukherjee, A.: Some principles for designing a wide-area WDM optical network. IEEE/ACM Trans. Netw. **4**(5), 684–696 (1996)
20. Prathombutr, P., Stach, J., Park, E.K.: An algorithm for traffic grooming in WDM optical mesh networks with multiple objectives. Telecommun. Syst. **28**(3–4), 369–386 (2005)
21. Raghavan, S., Stanojević, D.: Branch-and-price for WDM optical networks with no bifurcation of flow. INFORMS J. Comput. **23**(1), 56–74 (2011)
22. Ramaswami, R., Sivarajan, K.N.: Design of logical topologies for wavelength-routed optical networks. IEEE J. Sel. Areas Telecommun. **14**(5), 840–851 (1996)
23. Ramaswami, R., Sivarajan, K., Sasaki, G.: Optical Networks: A Practical Perspective, 3rd edn. Morgan Kaufmann, San Francisco (2009)
24. Runge, P., Bunge, C.-A., Petermann, K.: All-optical wavelength conversion with extinction ratio improvement of 100 gb/s rz-signals in ultralong bulk semiconductor optical amplifiers. IEEE J. Quantum Electron. **46**(6), 937–944 (2010)
25. Somani, A.: Survivability and Traffic Grooming in WDM Optical Networks. Cambridge University Press, Cambridge (2006)
26. Subramaniam, S., Azizoglu, M., Somani, A.: All-optical networks with sparse wavelength conversion. IEEE/ACM Trans. Netw. **4**(4), 544–557 (1996)
27. Sung, C.S., Song, S.H.: Branch-and-price algorithm for a combined problem of virtual path establishment and traffic packet routing in a layered communication network. J. Oper. Res. Soc. **54**(1), 72–82 (2003)
28. Talebi, N., Shahabadi, M.: All-optical wavelength converter based on a heterogeneously integrated GaP on a silicon-on-insulator waveguide. J. Opt. Soc. Am. B **27**(11), 2273–2278 (2010)
29. Vanderbeck, F.: Implementing mixed integer column generation. In: Desaulniers, G., Desrosiers, J., Solomon, M.M. (eds.) Column Generation. Kluwer (2005)
30. Vanderbeck, F., Wolsey, L.A.: An exact algorithm for IP column generation. Oper. Res. Lett. **19**, 151–159 (1996)
31. Zhu, K., Mukherjee, B.: Traffic grooming in an optical WDM mesh network. IEEE J. Sel. Areas Commun. **20**(1), 122–133 (2002)