WDM Optical Design using Branch-and-Price

S. Raghavan Daliborka Stanojević

The Robert H. Smith School of Business
University of Maryland
College Park, MD 20742-1815

Abstract

In this paper, we present an exact solution procedure for the design of two-layer WDM optical networks with wavelength changers and bifurcated flows. This design problem closely resembles traditional multicommodity flow problem, only, 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 the first branch-and-price algorithm for a general WDM optical network settings with no assumptions on the number of logical links that can be established between nodes in the network. We present the results of our 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.

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 simultaneous routing of traffic and logical paths (lightpaths) in the corresponding network layers. The two layers commonly considered in WDM optical network design include physical topology defined by an actual physical network of optical fibers, and the virtual (logical) topology defined by logical paths established over the physical topology.

The general 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 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 node equipment used in WDM optical networks may include transmitters and receivers, multiplexers and demultiplexers, amplifiers, wavelength changers, etc. However, for the purpose of the WDM network design problem studied in this paper it is sufficient to consider the presence of transmitters, receivers and wavelength changing devices only. Further, we assume that all transmitters and receivers are tunable to all wavelengths, and that all nodes in the network are equipped with the wavelength changers. This latter assumption is driven by the results of previous studies, which indicate that wavelength conversion may be desirable in WDM networks as a way of resolving equipment compatibility issues [4]. (More details on the tradeoffs between optical crossconnects with and without wavelength conversion capability can be found in Ramaswami and Sivarajan [15].) With regard to the restrictions imposed by the capacity of the optical fibers, usually two factors are considered - fiber bandwidth/capacity and the number of wavelengths/lightpaths that can be supported by a given fiber.

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 [4]), most often it is assumed either that the physical topology is given and fixed [12], [16], or, that only a part of the physical topology is fixed, and a part of it is subject to change. (For example, Hu and Leida [10], and Konda and Chou [11], 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. As such, there is no routing cost. 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 [15], 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 consider the situation where bifurcation of flow is not permitted (and is more appropriate for SONET circuits and networks that allow grooming of the higher speed connections) in a companion paper [14].

The general WDM optical network design problem is often seen as two interrelated problems, the logical (lightpath) topology design problem and the lightpath routing problem. The logical (lightpath) topology design problem is sometimes defined as the problem that is concerned with determining the number of lightpaths to be established between all pairs of nodes, and routing of the traffic over the established logical topology [15]. By this definition, the logical topology design (LTD) problem is not concerned with the actual routing of the lightpaths in the 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 known as the lightpath routing problem.

In this paper, we use different terminology for the subproblems of the WDM optical network design. Specifically, we use the term **logical topology design** 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.

Literature Review: Most of the previous research sequentially solves the WDM optical network design problem using the logical topology design problem and the lightpath routing problem. These heuristic procedures usually simplify the logical topology design 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 LTD. The heuristic approaches that first heuristically solve the LTD problem are usually simpler, since the problem of traffic routing 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 [16], where they proposed several heuristic procedures for LTD problem. The logical topology determined this way was then used as input for an MIP that solves the traffic routing problem over the fixed logical topology. The computational results in [16] indicate that the linear program based on the logical topology determined by one of the LTD procedures called HLDA provides the least congestion in most cases when compared to other heuristic LTD procedures proposed in the same paper.

Haque et al. [9] used a column generation based approach for the traffic routing over the fixed logical topology. Their procedure is a part of an algorithm that solves WDM optical network design problem in optical networks that do not have wavelength conversion abilities, impose a constraint on maximum allowable delay for each commodity, and allow at most one lightpath between any two nodes. Computational experiments in [9] 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 in up to 15% improvement in the network throughput.

An alternative heuristic approach for solving WDM optical network design problem that simplifies LTD problem by pre-specifying a small number of lightpaths that can be used for LTD was used by Prathombutr et al. [13]. In [13], the set of lightpaths that can be used for LTD 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 search for good solutions by defining chromosomes with feasible logical topology. The routing of traffic over the

determined 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 [20].

Banerjee and Mukherjee [4] proposed an 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 [4] included three networks: 14-node NSFNET network, 15-node PACBELL network, and 20-node randomly generated network. In all tests it was assumed that there is only one 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. Studies of Ball and Vakhutinsky [2, 3], for example, consider two-layer network design 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 the networks with and without fault tolerance to single link failures. Given the computationally hard nature of this problem, Ball and Vakhutinsky [2] proposed several improvement and simplification strategies, including variable aggregation, use of 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. [7] considered a general problem for the layered telecommunication networks. They studied a two-layer network design problem, where the network is defined by the physical network $N = (V, L)$, and the **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. It was also assumed that the demand of any given commodity can have one of the two possible values only. 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.

Recently, Belotti and Malucelli [6] 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 [6]). 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 proposed column generation starts with a subset of lower and higher layer paths and progressively solves the problem. If the solution found by the column generation is fractional a heuristic rounding algorithm is applied in search for a feasible integer solution. The column generation is then repeated using the modified reduced costs that favor use of physical edges not close to its 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 in [18], 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 [18] 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 [18] is based on an incorrect assertion that the unit integer multicommodity flow prob-

lem is polynomially solvable. This is not true in general, and it can be shown that this problem is NP-hard (see [8]) 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 global lightpaths that can be established between any two nodes in the 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 make a realistic assumption that lightpaths that can be established in the network have a limited capacity. 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).

The rest of the paper is organized as follows. In the second section, we propose two simple heuristic procedures that can be used to obtain fast upper bounds for this problem. The mathematical formulation for the problem studied in this paper is presented in Section 3. In Section 4, we propose a novel branch-and-price algorithm for the mathematical formulation presented in Section 3. The results of our computational study are presented in Section 5. These results indicate that our branch and price procedures provide solutions that are on average within 2.5% of optimality on networks with up to 20 nodes. In Section 6 we provide concluding remarks.

2 Upper Bounding Heuristics for WDM Optical Network Design

The two following heuristic procedures for the WDM optical network design problem represent extensions of the HLDA algorithm proposed by Ramaswami and Sivarajan [16]. The initial steps of the first heuristic procedure are identical to that of the original HLDA procedure. 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 procedure. We modify this step by trying to establish lightpaths that utilize

as few physical hops (edges) as possible. Finally, we deal with routing of all the traffic that could not be served by the direct lightpaths. This final step also differs from the original HLDA, which stops once the logical topology is identified. In [16] the traffic routing is determined using an LP, which solves 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.

2.1 ModHLDA Algorithm

ModHLDA algorithm works in three phases. First, commodities with the highest demand are served with 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, 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 no sufficient 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 insufficient number of 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). (Note: All lightpaths with positive remaining capacity are considered in finding the shortest path of a given commodity). If the path found has capacity lower than the demand of a given commodity, the network capacity is updated (i.e., subtract the capacity of the shortest path from the capacity of the lightpaths used), and the shortest path algorithm is solved again for the same commodity. The procedure is repeated until no new paths can be found for a given commodity.

2.2 LPModHLDA Algorithm

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

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

Subject to:

$$\sum_{i:(i,k,l)\in\Lambda} f^{(s,d)}_{(i,k,l)} - \sum_{j:(k,j,l)\in\Lambda} f^{(s,d)}_{(k,j,l)} = \begin{cases} 1 - H^{(s,d)} & \text{if } k = d \\ H^{(s,d)} - 1 & \text{if } k = s \\ 0 & \text{otherwise} \end{cases} \quad \forall (s,d) \in \Omega, k \in V \tag{2}$$

$$\sum_{(s,d)\in\Omega} T^{(s,d)} f^{(s,d)}_{(i,j,l)} \leq 1 \quad \forall \lambda_{(i,j,l)} \in \Lambda \tag{3}$$

$$f^{(s,d)}_{(i,j,l)} \in R^1_+ \quad \forall (i,j,l) \in \Lambda, (s,d) \in \Omega \tag{4}$$

$$H^{(s,d)} \in R^1_+ \quad \forall (s,d) \in \Omega \tag{5}$$

The variables in the above linear program include flow variables $f^{(s,d)}_{(i,j,l)}$, each indicating the amount of flow of the commodity(s,d) carried over the l^{th} 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 HLDA, and set of commodities. Note that, since the logical topology is fixed, we need only the flow balance constraints (2), and constraints on the capacities of the lightpaths (3).

3 Mathematical Formulation for the WDM Optical Network Design

One way to formulate WDM optical network design problem is to use arc-based variables for definition of the logical topology. These variables simply tell us the number of lightpaths between a pair of nodes, but do not provide information regarding their routes. We need to define a set of multicommodity flow variables that determine the routing of the logical topology variables (lightpaths) on the physical network. We also need to define a set of multicommodity flow variables that determine routing of traffic over the logical topology variables (an arc-based formulation for the WDM optical network design problem is shown in Appendix A). The problem with this type of formulation is that it becomes computationally intractable as the size of the network increases. Consequently, we develop a column generation based procedure for two layer 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 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. We also differentiate between the lightpaths with known paths in the physical topology as the **local lightpaths**. The lightpaths with unspecified paths in the physical layer (that is, where we only know origins and destinations of the lightpaths) will be referred to as **global lightpaths**.

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 between nodes s and 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, and the set Z represents the set of all possible 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 global 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 of a given origin and destination are routed over the physical topology). The number of lightpaths z used in a given solution is specified by the local 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 indicator variables, $f_p^{(s,d)}$ indicate whether a given flow path p is used to carry traffic demand for the commodity (s,d). In this formulation, flow paths are defined over the global lightpaths (i.e., these variables do not tell us the exact propagation path of a given commodity in the physical topology). Finally, the lost traffic indicator variables, $H^{(s,d)}$, indicate whether demand of commodity (s,d) is lost or satisfied.

MIP-PATH-GLOBAL Formulation

$$Min \sum_{(s,d)\in\Omega} T^{(s,d)} H^{(s,d)} \tag{6}$$

Subject to:

$$\sum_{j:(i,j)\in\Lambda} Y^{(i,j)} \leq \Delta_t^i \quad \forall i \in V \tag{7}$$

$$\sum_{i:(i,j)\in\Lambda} Y^{(i,j)} \leq \Delta_r^j \quad \forall j \in V \tag{8}$$

$$Y^{(i,j)} - \sum_{z:O(z)=i,D(z)=j} X_z^{(i,j)} = 0 \quad \forall (i,j) \in \Lambda \tag{9}$$

$$\sum_{(i,j)\in\Lambda,z:(l,m)\in z} X_z^{(i,j)} \leq L_{lm} \quad \forall (l,m) \in A \tag{10}$$

$$Y^{(i,j)} - \sum_{p:(i,j)\in p} f_p^{(s,d)} \geq 0 \quad \forall (i,j) \in \Lambda, (s,d) \in \Omega \tag{11}$$

$$Y^{(i,j)} - \sum_{(s,d)\in\Omega,p:(i,j)\in p} T^{(s,d)} f_p^{(s,d)} \geq 0 \quad \forall (i,j) \in \Lambda \tag{12}$$

$$\sum_{p\in P^{(s,d)}} f_p^{(s,d)} + H^{(s,d)} = 1 \quad \forall (s,d) \in \Omega \tag{13}$$

$$f_p^{(s,d)} \in R^1 \quad \forall p \in P^{(s,d)}, (s,d) \in \Omega \tag{14}$$

$$Y^{(i,j)} \in R_+^1 \quad \forall (i,j) \in \Lambda \tag{15}$$

$$X_z^{(i,j)} \in Z_+^1 \quad \forall z \in Z, (i,j) \in \Lambda \tag{16}$$

Constraints (7) and (8) in this formulation limit the out/in degree of any node to be not larger than the total number of transmitters/receivers. Constraint set (9) ensures that all global lightpaths are defined in the physical topology through adequate number of the local lightpaths in the physical topology. Since the local lightpath variables are defined as integers, these constraints also guarantee that $Y^{(i,j)}$ variables are integer as well, and can be therefore defined as real variables (constraint (15)). Constraint set (10) represents a limit on the number of lightpaths that can be established on any physical edge. Constraint set (11) ensures that the flow path of any given commodity (s, d) can use global lightpath (i, j) only if that lightpath is included in the logical topology. Constraints (12) are capacity constraints limiting total flow over all global lightpaths established between two nodes. Constraint set (13) ensures that either all demand for a given commodity is satisfied, or is entirely lost.

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 (7) through (10) is a more general version of the unit integer multicommodity flow problem not suitable for standard types of integer multicommodity flow strengthening cuts such as lifted cover inequalities. And, 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.

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

One issue related to the MIP-PATH-GLOBAL 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 local 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 this WDM optical network design problem without flow bifurcation and propose alternative branch-and-price procedures in [17, 14].

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 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 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 and Wolsey [19] for a more comprehensive review of other important issues).

The initialization is crucial for two reasons. First, in order to implement column generation it is necessary to start with a feasible initial solution (even if it is a very bad one). Second, we need to be careful with respect to our choice of initial variables as a good set of initial variables could significantly speed up the search for the optimal solution. But, when column generation is used in combination with branch-and-bound (these procedures are referred to as **branch-and-price** algorithms), even the start with a good (or even optimal) set of initial columns, does not guarantee that the search will terminate early. This is directly related to the strength of the linear relaxation of the master problem, an issue that can completely halt progress of any branch-and-price algorithm. So, as in the standard integer programming techniques based on the branch-and-bound method, it is still very important to start with a strong linear relaxation of the master model.

The subproblem structure has a great 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. For this reason, we should try to formulate the master problem in such a way that the calculation of the reduced cost or solving the subproblem is relatively easy.

In the case of branch-and-price algorithms, we also need to make sure that the branching decisions are such that the structure of the subproblem is not destroyed in the subsequent iterations.

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

One of the good properties of the MIP-PATH-GLOBAL formulation is that all global lightpath variables can be included in the master model at the beginning of the search, and as flow variables are related only to the global lightpath variables and can be priced out separately, the pricing part of the column generation algorithm can be performed in a straightforward manner. We describe the details of this procedure next.

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

- nonpositive a_i and b_j variables for constraints (7) and (8) respectively
- unrestricted in sign $g^{(i,j)}$ variables for constraint (9)
- nonpositive $d_{(l,m)}$ variables for constraint (10)
- nonpositive $r_{(i,j)}^{(s,d)}$ variables for constraint (11)
- nonnegative $v_{(i,j)}$ variables for constraint (12)
- unrestricted in sign $w^{(s,d)}$ variables for constraint (13)

The reduced cost of any $X_z^{(i,j)}$ 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:
$RC_p^{(s,d)} = \sum_{(i,j)\in p} (r_{(i,j)}^{(s,d)} + T^{(s,d)} v_{(i,j)}) - w^{(s,d)}$.
The new lightpaths that need to be added to the restricted master problem are identified by finding paths for each (i, j) that minimize the following expression:

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

$- \sum_{(l,m)\in z} d_{(l,m)}$.
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 greater than $-g^{(i,j)}$ the lightpath is added. Otherwise, no new 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:

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

or

$$\sum_{(i,j)\in\Lambda} (r_{(i,j)}^{(s,d)} + T^{(s,d)} v_{(i,j)}).$$

This can be solved as the shortest path problem on an auxiliary network where the cost of an arc (i, j) is defined by the expression $r_{(i,j)}^{(s,d)} + T^{(s,d)} v_{(i,j)}$. 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.

4.1.1 Branching Strategy

Although only local lightpath variables $X_z^{(i,j)}$ need to be defined as integers in the MIP-PATH-GLOBAL formulation, our branching strategy uses global lightpath variables as well. The branching is performed hierarchically from the higher to the lower level. In other words, we first branch on the global lightpath variables, and then on the local lightpath variables. For the global lightpath variables, variable dichotomy is used, while branching decisions for the local lightpath variables are more complicated.

In order to understand the problem related to branching on local lightpath variables, consider what would happen if we performed branching based on the variable dichotomy. Basically, for a given fractional variable $X_z^{(i,j)}$, and its current fractional value $X_z^{(i,j)*}$, we would create 2 new nodes by adding the restriction $X_z^{(i,j)} \leq \lfloor X_z^{(i,j)*} \rfloor$ on one branch, and the restriction $X_z^{(i,j)} > \lceil X_z^{(i,j)*} \rceil$ on the other branch. Enforcing the second restriction is not a problem, however, the first restriction requires that we do not generate any new lightpaths that would have the same propagation path as lightpath z corresponding to variable $X_z^{(i,j)}$. As noted by Barnhart et al. [5], 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 get the path 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 k-shortest path is not eligible for addition to the model. In this paper,

we propose a different branching strategy that indirectly forces all local lightpath variables to be integer. This strategy is based on observation that if the **sum** of all local lightpath variables with the same origin and destination is integer over every arc traversed in the physical layer then either:
a) all local lightpath variables are integer, or
b) all local lightpath variables do not have integer values, but the solution can be interpreted as integer.

The first case is obviously correct, since if all local lightpath variables are integer, it directly follows that **sum** of all local 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 obvious. First note that if the sum of all local lightpath variables with the same origin and destination is integer over all arcs in the network, and there are fractional lightpath variables, then it must mean that there is at least one pair of nodes with at least 4 fractional local lightpath variables that have that pair of nodes as their origin and destination. (If there was only one fractional lightpath variable, the sum of all local lightpath variables with the same origin and destination would not be integral. Similarly, if there were only two fractional lightpath variables, these lightpath variables would either have to follow the same path in the physical layer, which is not possible, or, the sum of all local lightpath variables over all the arcs in the network would not be integral. In other words, there must be at least 2 fractional lightpaths using given arc at the same time, or none. Since 2 lightpaths cannot have identical propagation path in the physical layer, it follows that we must have at least 4 fractional lightpath variables.) An example of situation where there are 4 fractional lightpath variables is shown in Figure 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 these 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 sum of all 'flows' going out of that node, it follows that in any fractional solution that satisfies requirement of integral sum of lightpath variables over all arcs, we can eliminate some of the fractional lightpaths so that integral solution is obtained, without any impact on other aspects of the current solution. In the example shown in Figure 1, we could, for example, eliminate lightpaths L_2 and L_4, and round up values of variables associated with variables L_1 and L_3, without making any other changes to current solution. This finding allows us to branch on the **sum** of all local lightpath variables with

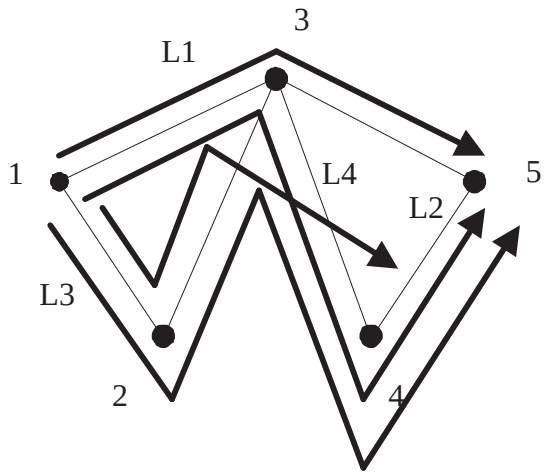

Figure 1: Example of branching issue for the MIP-PATH-GLOBAL formulation

the same origin and destination that are using same arc in the physical layer, instead of direct branching on individual local 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 global lightpath variables. If there are no such variables, go to Step 2, otherwise, select one fractional global variable, and create 2 new nodes using variable dichotomy.

Step 2. Check if there is an arc with a fractional sum of all local lightpath variables with the same origin and destination using that arc. If there is no such arc, go to Step 3, 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 local lightpath variables in the current solution.

Exit.

Finally, in order to ensure feasibility of the restricted master model at the beginning of the column generation procedure at each node of the branch-and-bound tree, we modify the MIP-PATH-GLOBAL formulation by using the following approach. Specifically, we add a high cost, artificial variable to the objective function, as well as in the constraint (13), which ensures that either all

demand for a given commodity is satisfied, or is entirely lost. We also add an artificial variable when we add constraints of the type $Y^{(i,j)} \geq \lceil Y^{(i,j)*} \rceil$ and $\sum_{z:(m,n)\in z} X_z^{(i,j)} \geq \lceil \sum_{z:(m,n)\in z} X_z^{(i,j)*} \rceil$ in our branching rules.

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

The branch-and-price algorithm for the MIP-PATH-GLOBAL formulation can be used with minor modifications for certain alternative network design objectives.

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

$Min \quad \sum_i (\Delta_t^i + \Delta_r^i)$

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 Computational Experiments

We coded the procedures presented in this paper in Microsoft Visual C++, ILOG *Maestro* [1] library, and CPLEX 9.0. All computations were performed on a workstation with 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. [13].

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 have 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 uniform distribution to determine commodities with a positive demand.)

The third and the fourth set of test instances are from the study of Parthombutr et al. [13]. These include a simple 6-node network and the NSFNET network. Since instances defined in [13] 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 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 capacity of a lightpath. As in [13], 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 Figures 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 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

	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

Table 1: Demand matrix for 6-node network example in Figure 2

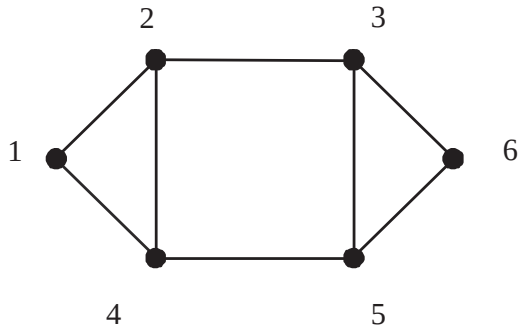

Figure 2: Physical topology of the 6-node optical network

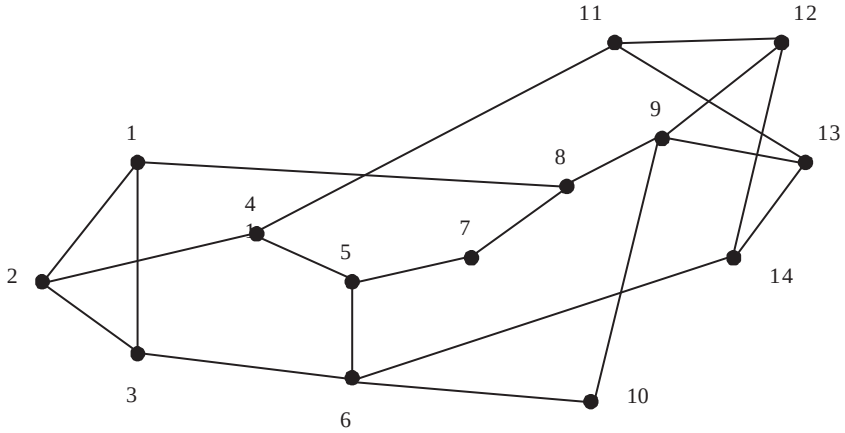

Figure 3: Physical topology of the NSFNET optical network

	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 2: Demand matrix for the NSFNET network shown in Figure 3

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 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 second CPU time limit.

Table 3 provides the results for the randomly defined instances that have 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 LPModHLDA are 3.54% better than those provided by ModHLDA, while the average CPU time for ModHLDA and LPModHLDA was 0.02 and 1.26 seconds respectively. Our branch-and-price algorithm, on the other hand, required 161.53 seconds 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 LPModHLDA 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 and upper bound at the end of the search was 1.78%[1]. In these five instances average improvement of the upper bound of our branch-and-price algorithm over the LPModHLDA was 2.6%.

[1]The percentage gaps in these tests were calculated as $\frac{UB-LB}{totaldemand-UB} * 100\%$. 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 provides the results for the randomly defined instances that have an incomplete graph in the physical layer. In this case, LPModHLDA provided upper bounds that were 0.62% better than those provided by ModHLDA. However, the upper bounds found by LPModHLDA 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 LPModHLDA was 0.11%.

The results for test instances defined in the study of Prathombutr et al. [13] 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 nodes network, and in one instance of the NSFNET network. In the 6 node network instances, the average optimality gap for LPModHLDA was 2.99%, while the average improvement of LPModHLDA over ModHLDA 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 needed to solve the LP at each node of the branch-and-bound tree. In these instances, LPModHLDA provided solutions that were on average 5.74% better than the ones provided by ModHLDA. Additionally, the solutions provided by LPModHLDA were better than the upper bounds found by our branch-and-price algorithm. The average improvement provided by LPModHLDA 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 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 does not represent an actual restriction and can be therefore relaxed.

Table 7 provides the summary of our computational tests with 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 lightpaths is relaxed.

6 Conclusion

The integrated design of logical topology and traffic grooming is an extremely challenging problem, and to our knowledge, for ease of computation, such problems have typically been broken up into two

			B&P Algorithm			ModHLDA		LP ModHLDA	
$\|V\|$	$\|C\|$	D	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

Table 3: Minimizing the lost traffic in the network. Complete physical network with 2 fibers (fiber capacity: 2 lightpaths) between all pairs of nodes.

				B&P Algorithm			ModHLDA		LP ModHLDA	
$\|V\|$	$\|A\|$	$\|C\|$	D	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

Table 4: Minimizing the lost traffic in the network. Incomplete physical network with 2 fibers (fiber capacity: 20 lightpaths) on all arcs.

		B&P Algorithm			ModHLDA		LP ModHLDA	
T/R Nbr	Wav. Nbr	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

Table 5: Minimizing the lost traffic in the network. The 6-node network with a single fiber on each arc.

T/R Nbr	Wav. Nbr	B&P Algorithm LB	UB	CPU (sec)	ModHLDA UB	CPU (sec)	LP ModHLDA 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

Table 6: Minimizing the lost traffic in the network. The NSFNET network with a single fiber on each arc.

Net. Type	Finite Wavelengths Avg. Gap	Avg. CPU (sec)	Infinite Wavelengths 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

Table 7: Summary: Minimizing the lost traffic in the network. Branch-and-Price results for WDM networks with and without constraint on the number of wavelengths in the network.

parts. In this paper we presented the first branch-and-price algorithm for the integrated design of logical topology and traffic routing in WDM optical networks with bifurcation of flow and a general network setting which does not explicitly impose a limit on the number of lightpaths that can be established between any two nodes in the network. We also proposed two simple upper bounding heuristics ModHLDA and LPModHLDA that can be used as fast upper bounding procedures for this problem.

The results of our computational experiments performed over a set of four different optical networks with 5 to 20 nodes indicate that our branch-and-price algorithm solves the WDM optical network design problem with the average optimality gap of 2.33% within the pre-specified 3600 second 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.

Finally, we note that one of the nice properties of our branch-and-price algorithm is that it can be easily adjusted for use in certain alternative WDM optical network settings with different

design objectives. Limited computational experiments suggest that for the alternative objective of minimizing the total number of transmitters and receivers used in the network our branch-and-price algorithm, when directly applied, does not perform as well as the situation where we wish to maximize the traffic routed. (The average percentage gap for our branch-and-price algorithm was 16.42% with the average CPU time of 2576.78 seconds. In the case of infinite number of wavelengths, the average percentage gap was 7.59% with the average CPU time of 2563.93 seconds.) This presents another direction for future work.

References

[1] *ILOG Branch & Price & Cut Shortest Path Optimizers Prototype Manual.* ILOG, France, 2003.

[2] M. O. Ball and A. Vakhutinsky. Fault tolerant virtual path layout: Optimization models. In Sanso B and P. Soriano, editors, *Telecommunications Network Planning*, pages 210–217. Kluwer Academic, Boston, MA, 1998.

[3] M. O. Ball and A. Vakhutinsky. Fault-tolerant virtual path layout in ATM networks. *INFORMS Journal on Computing*, 13(1):76–94, 2001.

[4] D. Banerjee and B. Mukherjee. Wavelength-routed optical networks: Linear formulation, resource budgeting tradeoffs, and a reconfiguration study. *IEEE/ACM Transactions on Networking*, 8(5):598–607, 2000.

[5] C. Barnhart, C. A. Hane, and P. H. Vance. Using branch-and-price-and-cut to solve origin-destination integer multicommodity flow problems. *Operations Research*, 48(2):318–326, 2000.

[6] P. Belotti and F. Malucelli. Multilayer network design: a row-column generation algorithm. *Proceedings of International Network Optimization Conference*, pages B2.422–B2.427, 2005.

[7] G. Dahl, A. Martin, and M. Stoer. Routing through virtual paths in layered telecommunication networks. *Operations Research*, 47(5):693–702, 1999.

[8] N. Garg, V. V. Vazirani, and M. Yannakakis. Primal-dual approximation algorithms for integral flow and multicut in trees. *Algorithmica*, 18:3–20, 1997.

[9] A. Haque, Y. P. Aneja, S. Bandyopadhyay, A. Jaekel, and A. Sengupta. Some studies on the logical topology design of large multi-hop optical networks. *Optical Networks Magazine*, July/August 2002.

[10] J. Q. Hu and B. Leida. Traffic grooming, routing, and wavelength assignment in optical WDM mesh networks. *manuscript available at http://people.bu.edu/hqiang/papers/grw.pdf*, 2002.

[11] V. R. Konda and T.Y. Chow. Algorithm for traffic grooming in optical networks to minimize the number of transceivers. *Proc. IEEE 2001 Workshop on High performance switching and Routing*, pages 218–221, May 2001.

[12] B. Mukherjee, D. Banerjee, S. Ramamurthy, and A. Mukherjee. Some principles for designing a wide-area WDM optical network. *IEEE/ACM Transactions on Networking*, 4(5):684–696, 1996.

[13] P. Prathombutr, J. Stach, and E.K. Park. An algorithm for traffic grooming in WDM optical mesh networks with multiple objectives. *Telecommunications Systems*, 28(3–4):369–386, 2005.

[14] S. Raghavan and D. Stanojević. Branch-and-price for WDM optical networks with no bifurcation of flow. *Working Paper*, 2006. Current version at http://www.glue.umd.edu/˜raghavan/WDMP2.pdf.

[15] R. Ramaswami and K. Sivarajan. *Optical Networks: A Practical Perspective*. Morgan Kaufmann, San Francisco, second edition, 2002.

[16] R. Ramaswami and K. N. Sivarajan. Design of logical topologies for wavelength-routed optical networks. *IEEE Journal on Selected Areas in Telecommunications*, 14(5):840–851, 1996.

[17] D. Stanojević. *Optimization of Contemporary Telecommunciations Networks: Generalized minimum spanning trees and WDM Optical Networks*. PhD thesis, University of Maryland, College Park, 2005.

[18] CS Sung and SH Song. Branch-and-price algorithm for a combined problem of virtual path establishment and traffic packet routing in a layered communication network. *Journal of the Operational Research Society*, (54):72–82, 2003.

[19] F. Vanderbeck and L. A. Wolsey. An exact algorithm for IP column generation. *Operations Research Letters*, 19:151–159, 1996.

[20] K. Zhu and B. Mukherjee. Traffic grooming in an optical WDM mesh network. *IEEE Journal on Selected Areas in Communications*, 20(1):122–133, 2002.

Appendix A **MIP-ARC-GLOBAL formulation**:

Variables:

$H^{(s,d)}$ - Amount of traffic of commodity (s,d) that is not served. These variables are referred to as **lost traffic** variables

$Y^{(i,j)}$ - Number of lightpaths with origin at node i and destination at node j included in the logical topology. These variables will be referred to as **global lightpath variables**, as they do not provide the information regarding exact propagation paths of the lightpaths in the physical topology

$X^{(i,j)}_{(l,m)}$ - Number of lightpaths with origin at node i and destination at node j, using physical arc (l,m).

$f^{(s,d)}_{(i,j)}$ - Amount of flow (traffic) of commodity (s,d) carried over the lightpaths with origin at node i and destination at node j

MIP-ARC-GLOBAL:

$$Min \sum_{(s,d)\in\Omega} T^{(s,d)} H^{(s,d)} \tag{17}$$

Subject to:

$$\sum_{j:(i,j)\in\Lambda} Y^{(i,j)} \leq \Delta_t^i \quad \forall i \in V \tag{18}$$

$$\sum_{i:(i,j)\in\Lambda} Y^{(i,j)} \leq \Delta_r^j \quad \forall j \in V \tag{19}$$

$$\sum_{(i,j)\in\Lambda} X^{(i,j)}_{(l,m)} \leq L_{lm} \quad \forall (l,m) \in A \tag{20}$$

$$\sum_{l:(l,k)\in A} X^{(i,j)}_{(l,k)} - \sum_{m:(k,m)\in A} X^{(i,j)}_{(k,m)} = \begin{cases} Y^{(i,j)} & \text{if } k = j \\ -Y^{(i,j)} & \text{if } k = i \\ 0 & \text{otherwise} \end{cases} \quad \forall (i,j) \in \Lambda, k \in V \tag{21}$$

$$\sum_{i:(i,k)\in\Lambda} f_{(i,k)}^{(s,d)} - \sum_{j:(k,j)\in\Lambda} f_{(k,j)}^{(s,d)} = \begin{cases} 1 - H^{(s,d)} & \text{if } k = d \\ H^{(s,d)} - 1 & \text{if } k = s \\ 0 & \text{otherwise} \end{cases} \quad \forall (s,d) \in \Omega, k \in V \tag{22}$$

$$Y^{(i,j)} - f_{(i,j)}^{(s,d)} \geq 0 \quad \forall (s,d) \in \Omega, (i,j) \in \Lambda \tag{23}$$

$$Y^{(i,j)} - \sum_{(s,d)\in\Omega} T^{(s,d)} f_{(i,j)}^{(s,d)} \geq 0 \quad \forall (i,j) \in \Lambda \tag{24}$$

$$f_{(i,j)}^{(s,d)} \in R_+^1 \quad \forall (i,j) \in \Lambda, (s,d) \in \Omega \tag{25}$$

$$H^{(s,d)} \in R_+^1 \quad \forall (s,d) \in \Omega \tag{26}$$

$$Y^{(i,j)} \in R_+^1 \quad \forall (i,j) \in \Lambda \tag{27}$$

$$X_{(l,m)}^{(i,j)} \in Z_+^1 \quad \forall (l,m) \in A, (i,j) \in \Lambda \tag{28}$$

The first expression in the above formulation is the objective function that minimizes the total lost traffic. Constraints (18) and (19) are degree constraints for each node. They ensure that the total number of lightpaths originating/terminating at any node is limited by the number of transmitters/receivers at that node. Constraint (20) provides a bound on the number of lightpaths that can be established over a single arc. Constraint (21) is the flow balance constraint for the lightpaths. This constraint also guarantees that if the $X_{(l,m)}^{(i,j)}$ variables are integer, then the $Y^{(i,j)}$ variables are integer as well. Constraint (22) is the flow balance constraint for the traffic routed over established lightpaths. Constraint (23) limits traffic to lightpaths established in the logical topology. Constraint (24) limits the total amount of flow that can be sent over any single lightpath. Note that constraint (23) is redundant given the presence of constraint (24), but it is used here to strengthen the formulation.

The MIP-ARC-GLOBAL formulation closely resembles the formulation presented by Banerjee and Mukherjee in [4]. However, Banerjee and Mukherjee also define additional constraints to ensure that the entire capacity of any given lightpath is not used, and they include constraints that limit propagation delay of the lightpaths in the physical topology. The objective function in their formulation was the minimization of the average logical hop distance.