An Optimization Approach to Routing and Wavelength Assignment in WDM All-Optical Mesh Networks without Wavelength Conversion

Kyungsik Lee, Kug Chang Kang, Taehan Lee, and Sungsoo Park

This paper considers a routing and wavelength assignment problem (RWAP) for the implementation of efficient Wavelength Division Multiplexing all-optical mesh networks without wavelength conversion. For a given physical network and required connections, the solution to the RWAP consists in how to select a suitable path and wavelength among the many possible choices for each connection so that no two paths using the same wavelength pass through the same link, while minimizing the number of required wavelengths. We introduce an integer programming formulation of the RWAP, which has an exponential number of variables, and propose an algorithm to solve it based on the column generation technique. The proposed algorithm can yield high quality solutions and tight lower bounds at the same time. Though the proposed algorithm cannot guarantee optimal solutions, computational results show that the algorithm yields provably good solutions within a reasonable time.

Manuscript received Sept. 21, 2001; revised Feb. 19, 2002.

Kyungsik Lee (phone: +82 42 860 6569, e-mail: leeks@etri.re.kr) and Kug Chang Kang (e-mail: kckang@etri.re.kr) are with ETRI, Daejeon, Korea.

Taehan Lee (e-mail: s_thlee@cais.kaist.ac.kr) and Sungsoo Park (e-mail: sspark@cais.kaist.ac.kr) are with KAIST, Daejeon, Korea.

I. INTRODUCTION

The emerging demand for integrated Internet applications requires a substantially higher bandwidth than that offered by current networks based on Electronic Time Division Multiplexing (ETDM) technology. For this reason, all-optical networks based on Wavelength Division Multiplexing (WDM) technology are considered promising for the realization of large bandwidth networks in the future [1], [2]. Traditional ETDM networks use an electrical signal form to switch traffic along routes and restore signal strength. These networks are not fully utilizing the bandwidth of optical fiber (about 10 THz [3]) because only one carrier having a specific frequency (wavelength) of light is used on an optical fiber to transmit data signals that can be modulated at a maximum bit rate of the order of 10 Gbps. The high bandwidth of optical fibers can be utilized through WDM technology by which distinct data signals may share an optical fiber, provided they are transmitted on carriers having different wavelengths [4]. WDM all-optical networks promise data transmission rates several orders of magnitude higher than current networks. The key to high speeds in these networks is to maintain the signal in optical form, thereby avoiding the prohibitive overhead of conversion to and from electrical form [4].

In this paper, we consider a routing and wavelength assignment problem (RWAP) for the efficient implementation of WDM all-optical mesh networks without wavelength conversion. Consider an undirected mesh network G, with the node set V and the link set E, a set of pairs of nodes K, and the number of required connections r_k for each pair of nodes $k \in K$.

A connection between a pair of nodes $k \in K$ is realized on G by establishing a path between the two nodes (o_k, t_k) and assigning a specific wavelength to it. We assume that each link consists of two counter-propagating unidirectional fibers, and all paths are bidirectional and carried by these fibers. The solution to the RWAP consists in how to realize all the required connections so that no two paths of the same wavelength share a common link, while minimizing the number of required wavelengths.

Practical limitations on transmission technology and optical devices restrict the number of available wavelengths on a fiber (WDM systems that can carry up to 40 wavelengths on a fiber are commercially available currently), so that a good solution to the RWAP is important to the efficient implementation of WDM all-optical networks. It is also crucial for network planning and design to determine the wavelength requirements for different traffic patterns.

Many researchers have tried to solve the RWAP. However, they have concentrated on the development of heuristic algorithms because the RWAP is NP-hard [5]. Their solution approaches include 1) choosing a path by some specified rule and then assigning a wavelength to it in a greedy manner [2], [4], [6]-[9], 2) decoupling the problem into a routing problem and a wavelength assignment problem [10], and 3) using meta-heuristics like genetic algorithms [2], [11]. All the above approaches essentially make the routing decisions and the wavelength assignment decisions in a sequential manner. Interested readers can refer to [12]-[14] for the review of various routing and wavelength assignment approaches. In particular, [12] provides an excellent survey of the previous studies in this area.

In this paper, we propose a unified approach to solve the RWAP based on integer programming methods. Section II introduces an integer programming formulation that yields tight lower bounds on the wavelength requirements both theoretically and computationally. Using the formulation, we devise an efficient algorithm to solve the RWAP based on the column generation technique [15]. Section III describes an efficient algorithm for the column generation, and we present our algorithm for the RWAP in section IV. Computational results are given in section V.

II. MATHEMATICAL FORMULATION

In this section, we present an integer programming formulation of the RWAP and the procedure to solve the linear programming relaxation of the formulation to get a lower bound on the optimal wavelength requirements.

To present an integer programming formulation of the RWAP, we define the concept of an *independent routing configuration*. A routing configuration c is represented by a nonnegative vector $a_c \in Z_+^{|K|}$, such that $a_{ck} \le r_k$, for all $k \in K$. A routing configuration c is independent if we can realize a_{ck} connections for all $k \in K$ on G simultaneously using only one wavelength. Let C be the set of all possible independent routing configurations. Then, the integer programming formulation of the RWAP is as follows:

$$
\begin{aligned}
\text{(MP)} \quad & \text{minimize} && \textstyle\sum_{c \in C} z_c \\
& \text{subject to} && \textstyle\sum_{c \in C} a_{ck} z_c \ge r_k, \forall k \in K \qquad (1) \\
& && z_c \ge 0 \ \text{ integer}, \forall c \in C.
\end{aligned}
$$

Each decision variable, $z_c = l$, if an independent routing configuration c is realized l times by using l different wavelengths, where l is a nonnegative integer. The objective function represents the number of required wavelengths. Constraints (1) ensure that all the required connections should be realized. We do not need to explicitly consider the assignment of wavelengths to paths in the formulation. After a solution to MP is obtained, we can assign wavelengths to the realized routing configurations.

A linear programming (LP) relaxation of a formulation can be obtained by dropping integrality restrictions imposed on the decision variables. Let MPL be the LP relaxation of MP, and let Z_{LP} be the optimal objective value of MPL. Clearly, $\lceil Z_{LP} \rceil$ is a lower bound on the optimal objective value of the RWAP. Our computational results, given in section V, show that the lower bounds are very tight. In addition, the tightness of the lower bounds can be shown theoretically. Interested readers can refer to Lee [5] for the details.

Generally, there are an exponential number of independent routing configurations for an instance of RWAP, that is, MP has an exponential number of decision variables. It is thus impractical to enumerate all the possible independent routing configurations to solve MPL with all the decision variables at hand. However, MPL can be solved efficiently by using the column generation technique of Gilmore and Gomory [15].

The outline of this method is as follows: We assume that a subset C' of C is given. Replacing C by C' in MPL yields the restricted linear program MPL', whose solutions are suboptimal to MPL. Then, we solve MPL' using the simplex method [16], a well-known method for solving linear programming problems, which yields an optimal solution to MPL' together with an optimal dual solution $\alpha_k^*, \forall k \in K$ associated with constraints (1). Now, using α_k^*, we search for a profitable independent routing configuration c' whose addition to MPL' may result in a decrease of the optimal objective value of MPL'. If there are no such independent routing configurations, the solution at hand is an optimal solution to MPL. Otherwise, we add c' to MPL', and then repeat the above process.

The key step of the above procedure is to check whether there is a profitable independent routing configuration. We can formalize this issue as another optimization problem, the so-called column (independent routing configuration) generation problem. Let $\alpha_k, \forall k \in K$ be the dual variables associated with constraints (1). Then the constraints in the dual of MPL' are

$$\sum_{k\in K} a_{ck}\alpha_k \le 1, \forall\, c \in C',$$
$$\alpha_k \ge 0, \forall\, k \in K.$$

Let $\alpha_k^*, \forall k \in K$ be an optimal solution to the dual of MPL'. Then it is optimal to the dual of MPL if and only if

$$\sum_{k\in K} a_{ck}\alpha_k^* \le 1, \forall\, c \in C \setminus C'.$$

Therefore, we may write the optimality condition for MPL as follows :

$$\max\{\sum_{k\in K} a_{ck}\alpha_k^* \mid c \in C\} \le 1\,. \tag{2}$$

Using condition (2), we can derive a formulation of the column generation problem as follows:

$$\begin{aligned}
\text{(SP) maximize } & \sum_{k\in K}\sum_{h\in P(k)} \alpha_k^* x_h \\
\text{subject to } & \sum_{h\in P(k)} x_h \le r_k, \forall k \in K \\
& \sum_{k\in K}\sum_{h\in P(k;e)} x_h \le 1, \forall e \in E \\
& 0 \le x_h \le 1 \text{ integer}, \forall h \in \textstyle\bigcup_{k\in K} P(k),
\end{aligned}$$

where $P(k)$ is the set of paths between o_k and t_k, for all $k \in K$, and $P(k;e)$ is the set of paths in $P(k)$ that pass through link e, for all $e \in E$ and $k \in K$. From the definition of the independent routing configuration, it can be easily shown that x is a feasible solution to SP if and only if a routing configuration c is independent, where $a_{ck} = \sum_{h\in P(k)} x_h$, for all $k \in K$. Now, suppose that we have an optimal solution to SP for a given optimal dual solution $\alpha_k^*, \forall k \in K$ to MPL'. By condition (2), if the optimal objective value of SP is greater than 1, the independent routing configuration corresponding to the solution is profitable and added to MPL', otherwise, no column is generated.

Using the above procedure, we do not need to enumerate all the possible independent routing configurations in advance of solving MPL. Now, we have only to be able to solve SP efficiently. Unfortunately, SP is NP-hard [5]. We devised, however, an optimization algorithm for SP based on the integer programming approach, which is presented in the next section.

III. ALGORITHM FOR THE COLUMN GENERATION PROBLEM FOR MPL

Consider the following integer program SP(d, u) :

$$\begin{aligned}
\text{SP}(d,u) \text{ maximize } & \sum_{k\in K}\sum_{h\in P(k)} d_k x_h \\
\text{subject to } & \sum_{h\in P(k)} x_h \le u_k, \forall k \in K \qquad (3)\\
& \sum_{k\in K}\sum_{h\in P(k;e)} x_h \le 1, \forall e \in E \qquad (4)\\
& 0 \le x_h \le 1 \text{ integer}, \forall h \in \textstyle\bigcup_{k\in K} P(k),
\end{aligned}$$

where $d \in R_+^{|K|}$ and $u \in Z_+^{|K|}$. For a given optimal solution α^* to the dual of MPL', SP(α^*, r) is the same as SP, which was presented in the previous section.

In this section, we present a branch-and-price algorithm to solve SP(d, u). SP(d, u) has an exponential number of path variables; however, its LP relaxation can also be solved efficiently by using the column generation technique [15] as in the case of MPL. In the following, we first present the column (path) generation procedure to solve the LP relaxation of SP(d, u). We then detail the branch-and-price algorithm for SP(d, u).

1. The Column Generation Procedure for SP(d, u)

To present the column generation procedure to solve the LP relaxation of SP(d, u), let SPL be the LP relaxation of SP(d, u). Let SPL' be the restricted linear program which can be obtained by replacing $P(k)$ in SPL by $P'(k) \subset P(k)$, for all $k \in K$. Let β_k, for all $k \in K$ and γ_e, for all $e \in E$ be the nonnegative dual variables associated with constraints (3) and (4), respectively. Then, for a given optimal dual solution (β^*, γ^*) to the dual of SPL', the reduced cost of each column (path) $h \in P(k)$ for each pair of nodes $k \in K$, c_{kh}, is

$$c_{kh} = d_k - \beta_k^* - \sum_{e\in E_h} \gamma_e^*,$$

where E_h is the set of links of which path h consists.

Then the column generation problem for each pair of nodes $k \in K$ is a shortest path problem between two nodes, o_k and t_k, of a pair of nodes $k \in K$ over a given network G with nonnegative links weights γ_e^*, for all $e \in E$. Denote the length of a shortest path p_k^* between a pair of nodes $k \in K$ as $L_{p_k^*}$. Then, if $d_k - \beta_k^* - L_{p_k^*} \le 0$, $\forall\, k \in K$, (β^*, γ^*) is also an optimal solution to the dual of SPL, so that there is no column whose addition to SPL' may result in a decrease of the optimal objective value of SPL'. Otherwise, the path p_k^* with $d_k - \beta_k^* - L_{p_k^*} > 0$ is added to SPL', for each $k \in K$.

Since the link weights are nonnegative, the column generation problems can be solved in polynomial time by using ordinary shortest path algorithms. We used Dijkstra's algorithm [17] for our implementation.

2. A Branch-and-Price Algorithm for SP(*d*, *u*)

The branch-and-price approach to solve integer programs is essentially the same as the branch-and-bound approach [16] except that the column generation is performed at every node of the branch-and-bound tree. For the general expositions of the branch-and-price approach, see Barnhart et al [18]. The following is the structure of our algorithm for SP(*d*, *u*) based on the branch-and-price approach.

Step 1: Initialization. Initialize SPL' with paths $P'(k) = \{h_k\}$, for each $k \in K$, where h_k is a shortest path between a pair of nodes $k \in K$ over the given network *G* with link weights equal to 1.

Step 2: SPL' solution. Solve SPL' using the simplex method. Let x^* be the obtained optimal solution to SPL', and let (β^*, γ^*) be the obtained optimal solution to the dual of SPL'.

Step 3: Column (Path) Generation. Use (β^*, γ^*) to solve the column generation problems (see sub-section 1). If one or more columns with positive reduced costs are generated, add these columns to SPL'. Otherwise, x^* is also an optimal solution to SPL, that is, we solved the LP relaxation of SP(*d*, *u*) at the root node of the branch-and-bound tree. Go to step 4.

Step 4: Integrality Test. If the obtained optimal solution x^* is integral, stop. We found an optimal solution to SP(*d*, *u*). Otherwise, go to step 5.

Step 5: Branch-and-price Procedure. Perform the branch-and-bound with the column generation to find an optimal solution to SP(*d*, *u*).

To initiate the branch-and-price algorithm for SP(*d*, *u*), we need to have $P'(k) \subset P(k)$, for all $k \in K$. To this end, we initialize $P'(k)$ as defined in step 1 of our algorithm. Note that $P'(k)$, for each $k \in K$, does not need to be initialized as such and can be an arbitrary subset of $P(k)$.

The key to developing the branch-and-price procedure in step 5 is identifying a branching rule that eliminates the current fractional optimal solution without making the column generation problem intractable after branching. To find an optimal solution to SP(*d*, *u*), we must maintain the ability to solve the column generation problem after branching. In the following, we present a branching rule that does not change the characteristics of the column generation problem after branching and present other details of our algorithm.

For a given optimal solution x^* to SPL, define $U(x^*, e) \subseteq K$, for all $e \in E$, such that $k \in U(x^*, e)$ if and only if $x_h^* > 0$ for some path $h \in P(k; e)$. Clearly, if x^* is integral, $|U(x^*, e)| \leq 1$, for all $e \in E$. Note that the converse is not always true. However, we can derive the following positive result.

Proposition 1. Suppose that an optimal solution x^* to SPL is given. If $|U(x^*, e)| \leq 1$, for all $e \in E$, x^* is either integral or there exists an integral feasible solution to SP(*d*, *u*) with the same objective value.

Proof. Suppose that $|U(x^*, e)| \leq 1$, for all $e \in E$, but x^* has fractional coordinates. For each $k \in K$, consider a graph $G_k = (V_k, E_k)$, where $V_k = V \cup \{s_k\}$, $(s_k, o_k) \in E_k$, and $e \in E_k$ if and only if $U(x^*, e) = \{k\}$, for each $e \in E$. To each link of G_k except $(s_k, o_k) \in E_k$, assign a link capacity of 1, and set the link capacity of $(s_k, o_k) \in E_k$ to u_k. Then, it is clear that, for each $k \in K$, the maximum flow value from s_k to t_k over G_k with link capacities defined as above is equal to $\sum_{h \in P(k)} x_h^*$. The maximum $s_k - t_k$ flow can be obtained by using ordinary maximum flow algorithms [17]. Further, for a given maximum flow solution, we can decompose flows into paths by using the flow decomposition algorithm [17]. Since given link capacities are all integral, the flow value of each path obtained by the flow decomposition algorithm is also integral. Therefore, we can obtain an integral feasible solution to SP(*d*, *u*) with the same objective value. □

By the above proposition, we have only to check if $|U(x^*, e)| \leq 1$, for all $e \in E$ instead of checking the integrality of the given optimal solution x^* to SPL in step 4. Now we present our branching rule. For the given optimal solution x^* to SPL, we first construct $U(x^*, e)$, for all $e \in E$. We then choose e^* with the lowest index such that $e^* = \arg\max_{e \in E} |U(x^*, e)|$. If $|U(x^*, e^*)| \leq 1$, we do not need to branch by proposition 1. Otherwise, we choose $k^* = \arg\max_{k \in U(x^*, e^*)} \sum_{h \in P(k; e^*)} x_h^*$ with the lowest index and divide $U(x^*, e^*)$ into two disjoint set $U_1(x^*, e^*)$ and $U_2(x^*, e^*)$, such that $U_1(x^*, e^*) = \{k^*\}$ and $U_2(x^*, e^*) = U(x^*, e^*) \setminus \{k^*\}$. Then we make two child nodes in the branch-and-bound tree such that each $k \in U_1(x^*, e^*)$ cannot use the link e^* in the first node and each $k \in U_2(x^*, e^*)$ cannot use the link e^* in the second node. That is, for the first node, we require

$$x_h = 0, \quad \text{for all} \quad h \in P(k; e^*) \quad \text{and} \quad k \in U_1(x^*, e^*),$$

and for the second node, we require

$$x_h = 0, \quad \text{for all} \quad h \in P(k; e^*) \quad \text{and} \quad k \in U_2(x^*, e^*) .$$

To satisfy the above requirements, for each path variable x_h that is already generated, we first set the upper bound of the variable to 0 if $h \in P(k; e^*)$ and $k \in U_1(x^*, e^*)$ at the first node. We perform a similar bound setting at the second node. Furthermore, at the first node, for each $k \in U_1(x^*, e^*)$, we perform the column generation procedure over the network obtained by removing e^* from the given network *G*. The same scheme is applied at the second node. Since column generation problems can be solved by using ordinary shortest path algorithms over the modified networks, they remain tractable after branching.

Finally, we used the best bound rule [16] as the node selection rule in the branch-and-bound tree at step 5.

IV. OVERVIEW OF THE ALGORITHM FOR THE RWAP

In this section, we present the algorithm to solve the RWAP. Our algorithm can be outlined as follows: We first perform a greedy heuristic procedure, which is devised by using the algorithm for SP(*d*, *u*) presented in section III, to get a feasible solution to MP. Recall that a subset *C*' of *C* is assumed to be given in the procedure to solve MPL. Therefore, the obtained heuristic solution provides not only an upper bound but also the columns to initiate the procedure to solve MPL. Then, we solve MPL using the procedure presented in section II. After the optimality of MPL is attained, if the obtained optimal solution to MPL is integral or the heuristic solution at hand can be shown to be an optimal solution to MP, then we are finished with an optimal solution to the RWAP. Otherwise, we go into the branch-and-bound phase with the current formulation of MPL. If the integral solution obtained in the branch-and-bound phase can be shown to be an optimal solution to MP, then we are finished. Otherwise, the fix-and-generated procedure is performed to try to find an improved solution.

The following is the overall structure of our algorithm to solve the RWAP.

Step 1: Greedy Heuristic Procedure

Step 1.1: Initialization. Set $Z_H = 0,\ C_H = \varnothing$.

Step 1.2: Find Configuration. Solve SP(**1**, *r*) by using the algorithm in section III, where **1** is a |*K*|-dimensional vector each of which elements is equal to 1. Let x^* be the obtained optimal solution, and let c^* be the corresponding independent routing configuration.

Step 1.3: Update Parameters. $C_H \leftarrow C_H \cup \{c^*\}$, $r_k \leftarrow (r_k - \sum_{h \in P(k)} x_h^*)$, $\forall k \in K$, and $Z_H \leftarrow Z_H + 1$.

Step 1.4: Termination. If $r_k > 0$, for some $k \in K$, go to Step 1.2.

Step 2: Lower Bounding Procedure

Step 2.1: MPL Solution. Solve MPL using the procedure in section II. Let z^* be the obtained optimal solution to MPL, and let Z_{LP} be the corresponding optimal objective value.

Step 2.2: Optimality Test. If z^* is integral, it is an optimal solution to MP, stop. Else if $Z_H = \lceil Z_{LP} \rceil$, the heuristic solution obtained in step 1 is an optimal solution to MP, stop.

Step 3: Branch-and-bound Procedure

Step 3.1: Integral Solution. Perform the branch-and-bound procedure [13] with the columns generated so far. Let $\bar{z}$ be the obtained integral solution, and let Z_B be the corresponding objective value.

Step 3.2: Optimality Test. If $Z_B = \lceil Z_{LP} \rceil$, $\bar{z}$ is an optimal solution to MP, stop.

Step 4: Fix-and-generate Procedure

Step 4.1: Variable Selection. Set $\hat{z} \leftarrow z^*$, and let *C*' be the set of columns generated so far. Choose a variable $\hat{z}_{c'}$ such that $c' = \arg\max_{c \in C'} \{\hat{z}_c - \lfloor \hat{z}_c \rfloor\}$ with the lowest index and has a fractional value.

Step 4.2: Fixing. Set $r_k = \max\{0, r_k - a_{c'k} \lceil \hat{z}_{c'} \rceil\}$ for all $k \in K$,

Step 4.3: Generation. Solve MPL with *C*' and the modified r_k's. Let z^* be the obtained optimal solution, and let Z_F be the corresponding objective value.

Step 4.4: Integrality Test. If z^* is integral, stop, otherwise, go to step 4.1.

Instead of the fix-and-generate procedure, we can consider a branch-and-price procedure to get an optimal solution to the RWAP. There are, however, some difficulties in devising a branch-and-price procedure using the formulation MP. The main reason is that it is difficult to devise a branching rule which does not destroy the structure of the column generation problem. For the detailed discussion, refer to Lee [5].

The fix-and-generated procedure cannot guarantee an optimal solution to the RWAP. However, computational results, given in the next section, show that the procedure can further improve the quality of solutions.

Before closing this section, we briefly explain how one can recover the set of paths used to route the required connections and the assignment of wavelengths to those paths from a solution z^* that is obtained by running the 4-step algorithm described above. First, note that an independent routing configuration is obtained by solving SP(*d*, *u*), so that the corresponding solution x^* to SP(*d*, *u*) represents the set of paths used to route connections realized by the configuration. Also, note that the paths can be assigned to one wavelength. Therefore, if we are to assign a specific wavelength to the configuration, we have only to assign the wavelength to those paths. Now, assume that $z_c^* = l$, where *l* is a positive integer. Then, we assign *l* different wavelengths to the corresponding configuration *c* one at a time to realize the configuration *l* times.

V. COMPUTATIONAL RESULTS

We applied our algorithm for the RWAP to randomly generated problem instances on each of the three networks that are shown in Figs. 1, 2, and 3. The first network (NET1) is the NSF network [5], [19], the second one (NET2) is the COST239 network [20], and the third one (NET3) is from Grover et al [21].

We randomly generated two classes of problem instances on each network. Each class consists of 20 problem instances. In each problem instance of the first class (CLASS1), the number of required connections for each pair of nodes is set to 1 with probability 0.5 and 2 with the same probability. In each problem instance of the second class (CLASS2), the number of required connections for each pair of nodes is drawn uniformly from the set of integers in the range [1], [10]. Note that the numbers of pairs of nodes of NET1, NET2, and NET3 are 91, 153, and 55, respectively.

We implemented our algorithm in C++ using CPLEX [22] callable mixed integer library for the linear program solution routine and the branch-and-bound routine. The tests were performed on a pentium PC (700MHz).

Computational results are summarized in Tables 1 through 6. In those tables, LB refers to the lower bound obtained at step 2

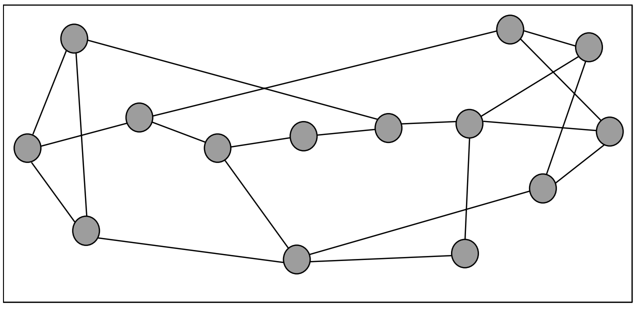

Fig. 1. The topology of NET1 (14 nodes, 21 links).

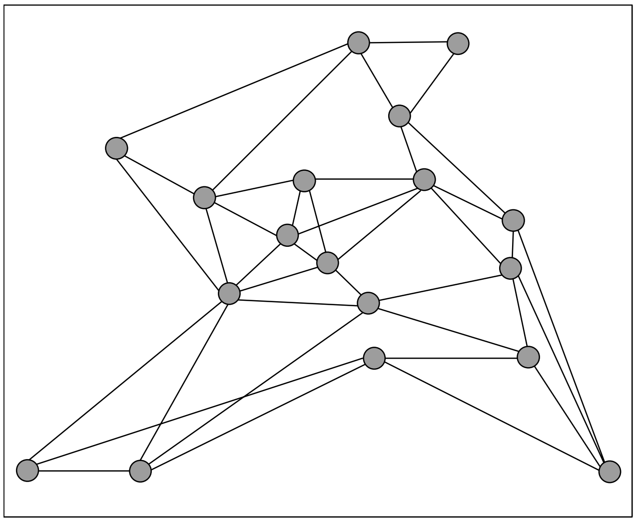

Fig. 2. The topology of NET2 (18 nodes, 39 links).

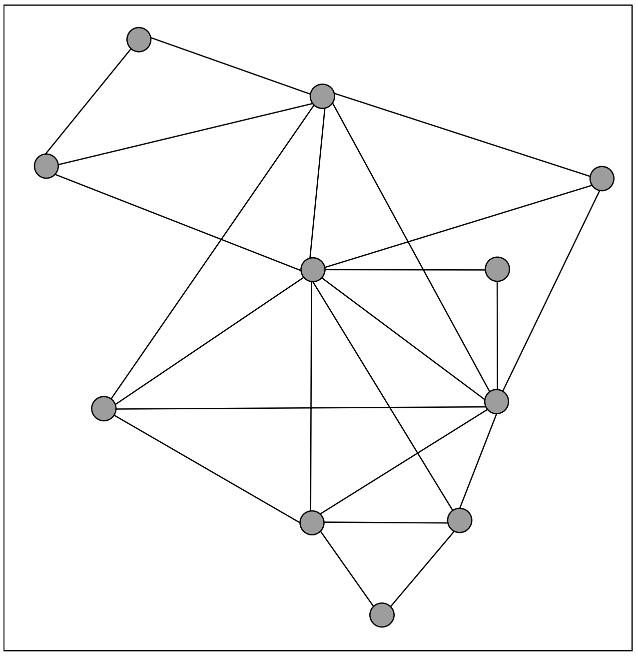

Fig. 3. The topology of NET3 (11 nodes, 23 links).

of our algorithm and equal to $\lceil Z_{LP} \rceil$. The headings Z_H, Z_B, and Z_F refer to the objective values of the integral solutions obtained by the greedy heuristic procedure, the branch-and-bound procedure, and the fix-and-generate procedure, respectively. Each number in parentheses in these columns represents the difference between the objective value of the integral solution obtained by the corresponding procedure and the corresponding lower bound. If the number is equal to 0, the solution is optimal. T_H , T_{LP} , T_B, and T_F refer to the amount of running time (seconds) it takes to perform step 1, 2, 3, and 4, respectively.

Tables 1 and 2 show the computational results on NET1. From the results for the problem instances of CLASS1 (Table 1), we can see that the lower bounds provided by the MPL are very tight. The greedy heuristic procedure yields sharp upper bounds, but it needs less than 1 second to get these solutions. The heuristic solutions were optimal in 11 problem instances, so that we did not need to perform step 3 and step 4. The branch-and-bound procedure could not yield improved solutions; however, we could get better solutions by the fix-and-generate procedure. In 14 out of 20 cases, the solutions obtained by our algorithm were optimal. The maximum difference between a lower bound and the solution obtained by our algorithm was 1. The maximum running time needed to perform our algorithm was less than 40 seconds.

From the results for CLASS2 (Table 2), we can also see that the lower bounds provided by our algorithm are tight. The differences between the solutions obtained by the greedy heuristic procedure and the corresponding lower bounds were relatively large in comparison to those in the case of CLASS1. However, the branch-and-bound procedure and the fix-and-generate procedure could yield improved solutions, so that the maximum difference between a solution obtained by our algorithm and the corresponding lower bound was 2. The differences were only 1 in 12 out of 20 cases.

Tables 3 and 4 show the computational results on NET2 for CLASS1 and CLASS2, respectively. For 9 problem instances of CLASS1, we could find optimal solutions. In each of the other cases, we could find a solution whose wavelength requirement is larger than the corresponding lower bound by only 1. The maximum running time needed to perform our algorithm was less than 160 seconds.

Table 4 reveals similar results to those on NET1. In this case, however, the branch-and-bound procedure reached the time limit (1800 seconds) before finding an integral solution. The maximum difference between a lower bound and the solution obtained by our algorithm was 3. The average difference was about 2. The maximum running time needed to perform our algorithm was reasonable.

Tables 5 and 6 show the computational results on NET3. The results confirm that our algorithm consistently provided tight lower bounds and near-optimal solutions within a short time.

Table 1. Summary of computational results on NET1 (CLASS1).

No	LB	Z_H	Z_B	Z_F	T_H	T_{LP}	T_B	T_F
1	18	19 (1)	19 (1)	19 (1)	0.41	3.08	1.01	2.09
2	18	19 (1)	19 (1)	18 (0)	0.66	5.20	1.96	10.32
3	18	18 (0)	-	-	0.45	3.48	-	-
4	19	19 (0)	-	-	0.37	2.80	-	-
5	20	21 (1)	21 (1)	21 (1)	0.35	2.53	0.68	5.37
6	20	20 (0)	-	-	0.44	2.89	-	-
7	19	19 (0)	-	-	0.41	3.13	-	-
8	18	20 (2)	20 (2)	19 (1)	0.36	3.37	1.54	8.41
9	20	20 (0)	-	-	0.40	3.04	-	-
10	19	20 (1)	20 (1)	19 (0)	0.36	2.83	1.21	6.45
11	19	19 (0)	-	-	0.41	4.21	-	-
12	20	20 (0)	-	-	0.59	3.02	-	-
13	18	19 (1)	19 (1)	19 (1)	0.44	4.09	1.16	10.01
14	17	19 (2)	19 (2)	18 (1)	0.37	4.02	1.44	8.34
15	18	19 (1)	19 (1)	18 (0)	0.41	4.32	1.37	8.41
16	18	18 (0)	-	-	0.29	3.32	-	-
17	19	19 (0)	-	-	0.48	4.55	-	-
18	19	19 (0)	-	-	0.38	2.80	-	-
19	18	19 (1)	19 (1)	18 (0)	0.37	4.00	1.85	33.13
20	18	18 (0)	-	-	0.41	3.26	-	-

Table 2. Summary of computational results on NET1 (CLASS2).

No	LB	Z_H	Z_B	Z_F	T_H	T_{LP}	T_B	T_F
1	73	79 (6)	74 (1)	73 (0)	1.85	2.51	18.46	19.69
2	64	71 (7)	66 (2)	65 (1)	1.26	2.66	94.60	22.19
3	65	66 (1)	66 (1)	65 (0)	1.11	2.22	19.63	16.75
4	63	66 (3)	66 (3)	65 (2)	1.26	2.80	48.45	16.20
5	71	75 (4)	72 (1)	72 (1)	1.63	2.21	507.48	14.86
6	58	62 (4)	60 (2)	60 (2)	1.25	3.41	139.04	17.60
7	67	69 (2)	69 (2)	68 (1)	1.37	2.11	71.55	15.80
8	73	79 (6)	75 (2)	75 (2)	1.60	2.24	120.44	19.59
9	68	74 (6)	70 (2)	69 (1)	1.66	3.46	203.94	26.24
10	59	67 (8)	61 (2)	60 (1)	1.06	3.48	232.40	19.42
11	67	73 (6)	69 (2)	68 (1)	1.24	3.71	1126.29	24.77
12	56	58 (2)	58 (2)	57 (1)	1.23	1.60	10.54	15.72
13	70	74 (4)	72 (2)	72 (2)	1.63	3.10	100.42	21.89
14	66	73 (7)	68 (2)	67 (1)	1.43	3.98	532.90	20.14
15	72	76 (4)	74 (2)	72 (0)	1.25	2.63	343.56	20.68
16	79	81 (2)	80 (1)	80 (1)	1.08	0.64	10.40	12.40
17	71	78 (7)	73 (2)	73 (2)	1.54	3.30	97.13	24.94
18	69	74 (5)	71 (2)	70 (1)	1.26	1.35	5.33	22.30
19	69	71 (2)	70 (1)	70 (1)	1.36	0.71	0.74	53.86
20	73	75 (3)	74 (1)	74 (1)	1.33	1.45	10.03	17.88

Table 3. Summary of computational results on NET2 (CLASS1).

No	LB	Z_H	Z_B	Z_F	T_H	T_{LP}	T_B	T_F
1	18	18 (0)	-	-	0.84	15.94	-	-
2	18	19 (1)	19 (1)	19 (1)	1.53	25.22	8.96	26.26
3	16	17 (1)	17 (1)	17 (1)	0.96	23.35	9.31	35.08
4	17	17 (0)	-	-	1.04	33.82	-	-
5	18	18 (0)	-	-	1.02	35.66	-	-
6	17	18 (1)	18 (1)	18 (1)	0.96	28.57	10.98	64.45
7	17	18 (1)	18 (1)	18 (1)	1.03	31.98	12.14	74.32
8	18	19 (1)	19 (1)	19 (1)	1.26	33.38	20.12	86.08
9	18	18 (0)	-	-	1.17	13.42	-	-
10	18	18 (0)	-	-	1.48	47.60	-	-
11	18	19 (1)	19 (1)	19 (1)	1.21	33.80	14.04	102.4
12	18	18 (0)	-	-	1.19	17.51	-	-
13	18	19 (1)	19 (1)	19 (1)	0.91	18.56	5.02	35.91
14	17	17 (0)	-	-	0.96	30.37	-	-
15	18	18 (0)	-	-	1.04	38.50	-	-
16	18	19 (1)	19 (1)	19 (1)	0.97	28.89	12.62	73.18
17	19	19 (0)	-	-	1.16	30.53	-	-
18	18	19 (1)	19 (1)	19 (1)	1.21	31.70	13.57	53.48
19	17	18 (1)	18 (1)	18 (1)	1.47	36.35	16.97	80.53
20	18	19 (1)	19 (1)	19 (1)	0.96	24.56	11.88	65.89

Table 4. Summary of computational results on NET2 (CLASS2).

No	LB	Z_H	Z_B	Z_F	T_H	T_{LP}	T_B	T_F
1	62	65 (2)	*	63 (1)	4.31	28.63	1800.0	180.06
2	65	69 (4)	*	66 (1)	3.44	28.03	1800.0	144.56
3	66	70 (4)	*	67 (1)	12.64	102.00	1800.0	123.32
4	67	71 (4)	*	70 (3)	3.41	27.43	1800.0	181.53
5	58	62 (4)	*	60 (2)	2.77	31.20	1800.0	137.65
6	73	78 (5)	*	76 (3)	3.34	9.64	1800.0	65.77
7	65	69 (4)	*	67 (2)	2.90	21.93	1800.0	118.90
8	60	63 (3)	*	62 (2)	4.15	21.53	1800.0	103.82
9	67	71 (4)	*	69 (2)	2.92	24.59	1800.0	126.81
10	62	65 (3)	*	64 (2)	3.99	23.53	1800.0	140.27
11	61	64 (3)	*	63 (2)	2.99	24.83	1800.0	105.21
12	70	74 (4)	*	73 (3)	1.01	28.72	1800.0	114.46
13	69	76 (7)	*	71 (2)	3.56	29.05	1800.0	143.31
14	67	74 (7)	*	70 (3)	3.49	26.80	1800.0	155.37
15	72	76 (4)	*	74 (2)	3.56	21.99	1800.0	133.81
16	63	67 (4)	*	65 (2)	3.02	23.27	1800.0	114.06
17	68	73 (5)	*	71 (3)	3.51	32.52	1800.0	155.84
18	61	66 (5)	*	63 (2)	2.56	20.14	1800.0	110.51
19	62	65 (3)	*	64 (2)	3.59	30.78	1800.0	99.47
20	64	67 (3)	*	65 (1)	4.05	19.11	1800.0	94.82

* an integral solution could not be found within 30 minutes (1800 seconds).

Table 5. Summary of computational results on NET3 (CLASS1).

No	LB	Z_H	Z_B	Z_F	T_H	T_{LP}	T_B	T_F
1	10	10 (0)	-	-	0.11	0.98	-	-
2	10	10 (0)	-	-	0.13	0.68	-	-
3	9	9 (0)	-	-	0.11	0.53	-	-
4	9	9 (0)	-	-	0.12	0.42	-	-
5	9	9 (0)	-	-	0.14	0.48	-	-
6	9	10 (1)	10 (1)	10 (1)	0.19	1.13	0.18	1.28
7	8	8 (0)	-	-	0.14	0.63	-	-
8	10	10 (0)	-	-	0.11	0.55	-	-
9	9	10 (1)	10 (1)	10 (1)	0.16	1.07	0.23	1.15
10	9	10 (1)	10 (1)	10 (1)	0.21	1.15	0.17	3.69
11	10	11 (1)	11 (1)	10 (0)	0.13	0.31	0.02	0.42
12	10	10 (0)	-	-	0.12	1.00	-	-
13	8	9 (1)	9 (1)	8 (0)	0.09	0.34	0.05	0.57
14	9	9 (0)	-	-	0.09	0.55	-	-
15	10	10 (0)	-	-	0.14	0.66	-	-
16	9	10 (1)	10 (1)	10 (1)	0.11	1.88	0.40	2.74
17	9	9 (0)	-	-	0.16	0.80	-	-
18	11	11 (0)	-	-	0.17	0.42	-	-
19	9	9 (0)	-	-	0.16	0.95	-	-
20	10	10 (0)	-	-	0.14	0.55	-	-

Table 6. Summary of computational results on NET3 (CLASS2).

No	LB	Z_H	Z_B	Z_F	T_H	T_{LP}	T_B	T_F
1	28	29 (1)	29 (1)	29 (1)	0.45	0.61	0.11	2.05
2	30	30 (0)	-	-	0.44	0.33	-	-
3	42	44 (2)	42 (0)	-	0.45	0.45	0.8	-
4	38	39 (1)	39 (1)	38 (0)	0.37	0.30	0.09	1.82
5	34	36 (2)	36 (2)	35 (1)	0.63	0.64	0.51	4.58
6	31	31 (0)	-	-	0.42	0.54	-	-
7	34	34 (0)	-	-	0.32	0.14	-	-
8	36	36 (0)	-	-	0.52	0.29	-	-
9	42	44 (2)	42 (0)	-	0.46	0.18	0.04	-
10	30	31 (1)	31 (1)	31 (1)	0.46	0.74	0.12	3.97
11	36	36 (0)	-	-	0.49	0.23	-	-
12	31	34 (3)	33 (2)	32 (1)	0.49	1.16	0.63	3.82
13	28	30 (2)	30 (2)	29 (1)	0.49	1.11	1.11	4.30
14	34	36 (2)	35 (1)	35 (1)	0.26	0.42	0.11	1.50
15	38	39 (1)	38 (0)	-	0.50	0.09	0.04	-
16	29	31 (2)	30 (1)	30 (1)	0.44	1.09	0.64	2.48
17	36	36 (0)	-	-	0.38	0.07	-	-
18	33	33 (0)	-	-	0.41	0.04	-	-
19	34	37 (3)	35 (1)	35 (1)	0.54	0.56	0.11	2.88
20	36	38 (2)	37 (1)	37 (1)	0.53	0.38	0.08	2.28

From the computational results presented so far, we can see that the lower bounds provided by MPL are very tight. We can also see that the greedy heuristic procedure (step 1) of our algorithm performs very well for problem instances in which the number of required connections is relatively small (CLASS1), so that the subsequent procedures (step 3 and 4) could not have many chances to yield better solutions. For the problem instances of CLASS2, the differences between the solutions obtained by the greedy heuristic solution and the lower bounds were relatively large. The branch-and-bound procedure and the fix-and-generate procedure, however, could reduce the differences and yield near-optimal solutions within a reasonable time.

VI. CONCLUSION

We have presented an efficient algorithm, based on the column generation technique, which provides high quality solutions to the routing and wavelength assignment problem (RWAP) for the implementation of efficient WDM all-optical mesh networks without wavelength conversion. In comparison with the heuristic algorithms proposed by the earlier studies in the literature, our algorithm gives a unified approach to routing and wavelength assignment rather than handling the routing decision and the wavelength assignment decision in a sequential manner. Computational results performed on the three networks show that the proposed algorithm can yield high quality solutions and tight lower bounds at the same time. Though the proposed algorithm cannot guarantee optimal solutions, computational results show that the algorithm yields *provably* good solutions within a reasonable time.

REFERENCES

[1] T.W. Um, J.K. Choi, Y.A. Kim, H.H. Lee, H.W. Jung, and S.G. Jong, “Signaling and Control Procedures Using Generalized MPLS Protocol for IP over Optical Network,” *ETRI J.*, vol. 24, no. 2, 2002, pp.69-80.

[2] L. Wuttisittikulkij and M.J. O’Mahony, “Design of Efficient and Practical Algorithm for Wavelength Assignment in Multi-wavelength Ring Transport Networks,” *Proc. of IEEE GLOBECOM*, Nov. 1997, pp.571-575.

[3] M. Settembre and F. Matera, “All Optical Implementations of High Capacity TDMA Networks,” *Fiber and Integrated Optics*,

vol.12, 1993, pp.173-186.

[4] P. Raghavan and E. Upfal, "Efficient Routing in All-Optical Networks," *Proc. of ACM STOC'94*, 1994, pp.134-143.

[5] K. Lee, *Routing and Capacity Assignment Models and Algorithms for the Design of Telecommunication Networks*, Unpublished Ph.D Dissertation, KAIST, Daejeon, Korea, 1998.

[6] C. Chen and S. Banerjee, "A New Model for Optimal Routing and Wavelength Assignment in Wavelength Division Multiplexed Optical Networks," *Proc.of IEEE INFOCOM*, Apr. 1996, pp.164-171.

[7] I. Chlamtac, A. Ganz, and G. Karmi, "Lightpath Communications: An Approach to High Bandwidth Optical WAN's," *IEEE Trans. on Comm*., vol.40, 1992, pp.1171-1182.

[8] K.C. Lee and V.O.K. Li, "A Wavelength Routing Algorithm in Wide-Area All-Optical Networks," *J.of Lightwave Technology*, vol. 14, no. 6, 1996, pp.1218-1229.

[9] R. Ramaswami and K.N. Sivarajan, "Routing and Wavelength Assignment in All-Optical Networks," *IEEE/ACM Tran.on Networking*, vol. 3, 1995, pp.489-500.

[10] D. Banerjee and B. Mukherjee, "Practical Approaches for Routing and Wavelength Assignment in Large All Optical Wavelength Routed Networks," *Proc.of IEEE INFOCOM*, June 1994, pp.110-119.

[11] L.G. Tan and M.C. Sinclair, "Wavelength Assignment between the Central Nodes of the COST239 European Optical Network," *Performance Engineering of Computer and Telecommunication Systems*, M. Merabti, M. Carew, and F. Ball eds., Springer, 1995, pp.235-247.

[12] H. Zang, J. Jue, and B. Mukherjee, "A Review of Routing and Wavelength Assignment Approaches for Wavelength-Routed Optical WDM Networks," *Optical Networks Mag*., vol.1, no.1, 2000, pp.47-60.

[13] D. Banerjee and B. Mukherjee, "Practical Approach for Routing and Wavelength Assignment in Large Wavelength-Routed Optical Networks," *IEEE J. on Selected Areas on Comm*., vol.14, 1996, pp.903-908.

[14] M. Alanyali and E. Ayanoglu, "Provisioning Algorthms for WDM Optical Networks," *IEEE/ACM Trans. on Networking*, vol. 7, 1999, pp.767-778.

[15] P.C. Gilmore and R.E. Gomory, "A Linear Programming Approach to the Cutting Stock Problem," *Operations Research*, vol. 9, 1961, pp.849-859.

[16] G.L. Nemhauser and L.A. Wolsey, *Integer and Combinatorial Optimization*, Wiley & Sons, 1998.

[17] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Prentice-Hall, 1993.

[18] C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, and P.H. Vance, "Branch-and-Price: Column Generation for Solving Huge Integer Programs," *Mathematical Programming: State of the Art*, J.R. Birge and K.G. Murty eds., University of Michigan, 1994, pp.186-207.

[19] J.H. Lee, "Design and Configuration of Reconfigurable ATM Networks with Unreliable Links," *ETRI J*., vol. 21, no.4, 1999, pp.9-22.

[20] B. Van Caenegem, W. Van Parys, F. De Turck, and P. Demesster, "Dimensioning of Survivable WDM Networks," *IEEE J.on Selected Areas in Comm*., vol. 16, 1998, pp.1146-1157.

[21] W. Grover, T. Bilobeau, and B. Venables, "Near Optimal Spare Capacity Planning in a Mesh," *Proc. of IEEE GLOBECOM '91*, 1991, pp.2007-2012.

[22] ILOG Inc., *ILOG CPLEX 6.5*, ILOG Inc., 1999.

Kyungsik Lee received a BS degree from the Department of Industrial Engineering, Seoul National University, Seoul, Korea, in 1993 and MS and PhD degrees in industrial engineering from the Korea Advanced Institute of Science and Technology, Daejeon, Korea, in 1995 and 1998, respectively. Since 1999, he has been working as a Senior Researcher in the Postal Technology Research Center, Electronics and Telecommunication Research Institute, Daejeon, Korea. His research interests include the theory of integer programming and combinatorial optimization, telecommunication network planning, supply chain management, logistics system optimization, and intelligent logistics information systems.

Kug Chang Kang received BS, MS and PhD degrees from the Department of Industrial Engineering, Seoul National University, Seoul, Korea, in 1990, 1992 and 1997, respectively. Since 1997, he has been working as a Senior Researcher in the Network Technology Laboratories, Electronics and Telecommunication Research Institute, Daejeon, Korea. His research interests include telecommunication network system engineering, network performance and quality, and network planning and design.

Taehan Lee received his BS degree in economics from Yonsei University, Seoul, Korea, in 1993, and his MS and PhD degrees in industrial engineering from the Korea Advanced Institute of Science and Technology, Daejeon, Korea, in 1997 and 2002, respectively. His research interests include the theory of integer programming and combinatorial optimization, telecommunication network planning and design, and logistics information systems.

Sungsoo Park received BS and MS degrees in industrial engineering from Seoul National University in 1980 and 1982, and his PhD degree in Operations Research from the Cornell University in 1989. Since 1990, he has been with the Department of Industrial Engineering at the Korea Advanced Institute of Science and Technology, Daejeon, Korea. His research interests include discrete optimization, telecommunication network planning and design, mathematical models in production and logistics systems.