Converter Placement Supporting Broadcast in WDM Optical Networks

Lu Ruan, Dingzhu Du, Xiaodong Hu, *Member*, *IEEE Computer Society*,
Xiaohua Jia, *Senior Member*, *IEEE*, Deying Li, and Zheng Sun

Abstract—Given a WDM optical network with wavelength channels on its fiber links, we consider the problem of finding the minimum set of network nodes such that, with wavelength converters at these nodes, broadcast can be supported in the network. We call this problem the Converter Placement problem. We model a given network using a graph G with colors on its edges and give a mathematical formulation for the problem based on the graph model. Two related problems, Color-Covering and Vertex Color-Covering, are given and analyzed. Both of them are shown to have a polynomial-time approximation with performance ratio $\ln n + 1$ and $\ln n$ is the best possible performance ratio unless $NP \subset DTIME(n^{poly \log n})$, where n is the number of vertices in G. Using these results, we show that the Converter Placement problem has a polynomial-time approximation with performance ratio $2(\ln n + 1)$ and $\frac{1}{2}\ln n$ is the best possible performance ratio unless $NP \subset DTIME(n^{poly \log n})$. We present an approximation algorithm to solve the Converter Placement problem and study the performance of the algorithm on randomly generated network topologies.

Index Terms—Network optimization, optical networks, WDM, converter placement, Color-Covering, Vertex Color-Covering.

1 Introduction

Optical networks employing wavelength division multiplexing (WDM) are believed to be the next generation networks that can meet the ever-increasing bandwidth demand of end users [7]. WDM provides an efficient way to utilize the tremendous bandwidth of a fiber (up to 50THz) [1]. In WDM, the optical spectrum is divided into many nonoverlapping channels, each corresponding to a different wavelength. Multiple users are multiplexed onto a fiber by using different wavelength channels.

A WDM optical network consists of optical routing nodes interconnected by fiber links. Each fiber link supports a set of wavelength channels and each routing node is able to route a signal coming in on one wavelength at an input port to any other output port. In WDM optical networks, one-to-one connections are supported by lightpaths. A lightpath consists of a physical path in the network between two end nodes and a wavelength channel assigned on each link along the path. If no wavelength converters are present, then all the links along the path must use the same wavelength. If a converter is available at node i, then a lightpath entering i using wavelength λ_1 can leave it using a different wavelength λ_2. It has been shown that placing wavelength converters at routing nodes can reduce the blocking probability experienced by connection requests [4], [5]. Wavelength conversion can be done either electronically or optically [6]. In the electronic approach, O-E-O conversion is required and it causes long delays. In all optical approaches, the optical signal is allowed in the optical domain throughout the conversion process. This kind of converter is costly at present. Therefore, in either case, it is desirable to minimize the number of wavelength converters used to save the hardware cost and conversion delay. Algorithms for placing a given number of wavelength converters in the network to minimize blocking probability were proposed in [13], [14], [15]. All these works assume that only one-to-one connections are present in the network. Recently, Sahasrabuddhe and Mukherjee introduced the concept of lighttree to support one-to-many (or multicast) connections in optical domain [8]. Unlike lightpath, lighttree connects a source to multiple destinations. At the branching points in the tree, lights are split to reach multiple outgoing links. In this way, multicast can be realized in the optical domain. To establish a lighttree, we first need to find a set of links that form the physical route of the tree and then assign a wavelength channel to each link on the tree. When two adjacent links use different wavelengths, wavelength conversion is needed at their joint node.

Multicast is increasingly popular on the Internet. A number of applications, such as video distribution and teleconferencing, require a multicast connection to be established. Some recent works studied wavelength requirements of multicast and the construction of multicast routing trees in optical networks [9], [10], [11], [12]. Broadcast is a special case of multicast where a source node connects to all the other nodes in the network. In order to support broadcast, a spanning lighttree needs to be

- *L. Ruan is with the Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455. E-mail: ruan@cs.umn.edu.*
- *D. Du is with the Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455 and the Institute of Applied Mathematics, Chinese Academy of Science, Beijing, China. E-mail: dzd@cs.umn.edu.*
- *X. Hu is with the Institute of Applied Mathematics, Chinese Academy of Science, Beijing, China.*
- *X. Jia and D. Li are with the Department of Computer Science, City University of Hong Kong, Kowloon Tong, Hong Kong.*
- *Z. Sun is with the School of Mathematics and System Sciences, Shandong University, Jinan, China.*

Manuscript received 3 Jan. 2000; revised 13 Sept. 2000; accepted 23 Feb. 2001. For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number 111335.

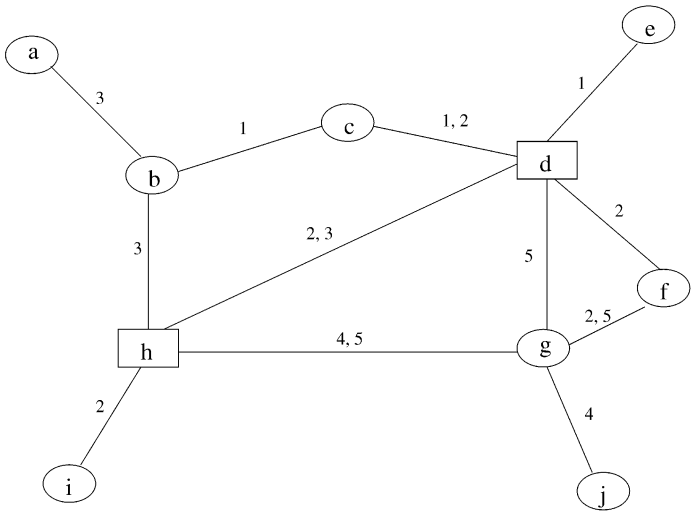

Fig. 1. The graph model of a network. The integers on each edge represent the colors on that edge.

constructed in the given network and each link in the lighttree occupies a wavelength channel. In this paper, we study how to find the minimum set of network nodes such that, with converters at these nodes, broadcast can be supported in the network or, equivalently, a spanning lighttree can be built using these nodes to do wavelength conversions. We call this problem the Converter Placement problem.

To tackle the Converter Placement problem, we introduce a graph model to represent the WDM optical networks and give a mathematical formulation for the problem using this graph model. We study the hardness of two related problems, namely Color-Covering and Vertex Color-Covering. Based on the results, we derive both a lower bound and an upper bound of the approximation of the Converter Placement problem and propose a greedy algorithm to generate approximation solutions to the problem.

The rest of the paper is organized as follows: Section 2 introduces the graph model and some definitions. Section 3 gives the mathematical formulations of the three problems. In Section 4, 5, and 6, we study Color-Covering, Vertex Color-Covering, and Converter Placement, respectively. In addition, two greedy algorithms are given in Section 4 and Section 6 for solving the Color-Covering problem and the Converter Placement problem. We give numerical results on the performance of the algorithm for the Converter Placement problem in Section 7 and Section 8 concludes the paper.

2 Network Model and Definitions

The network under our consideration is a set of routing nodes interconnected by fiber links. Each fiber link has a set of wavelength channels. We assume that each link is bidirectional and consists of a pair of fibers carrying information in opposite directions. Our goal is to find the minimum number of nodes to place converters such that broadcast can be supported.

We model the network using a connected graph $G = (V, E)$ and a mapping c from E to a set of colors represented by positive integers. The vertex set V of the graph represents the set of routing nodes and the edge set E of the graph represents the set of fiber links in the network. The colors on an edge represent the wavelength channels on that link. c is a function that assigns a set of colors to each edge $e \in E$.

Fig. 1 shows an example network modeled as a graph with colors on its edges. In this example, we can place converters at nodes d and h to support broadcast. A spanning lighttree with wavelength conversion at d and h is shown in Fig. 2.

Let G be a graph representing a given network. We introduce some definitions on G in the following.

For an edge e, $c(e)$ is the set of colors on edge e and we call it *edge color set* of e. For example, in Fig. 1, $c((c,d)) = \{1,2\}$ is the set of colors on edge (c,d).

For a vertex x, define the *vertex color set* of x, denoted by $vc(x)$, to be the union of the color sets of the edges incident to x. For example, in Fig. 1,

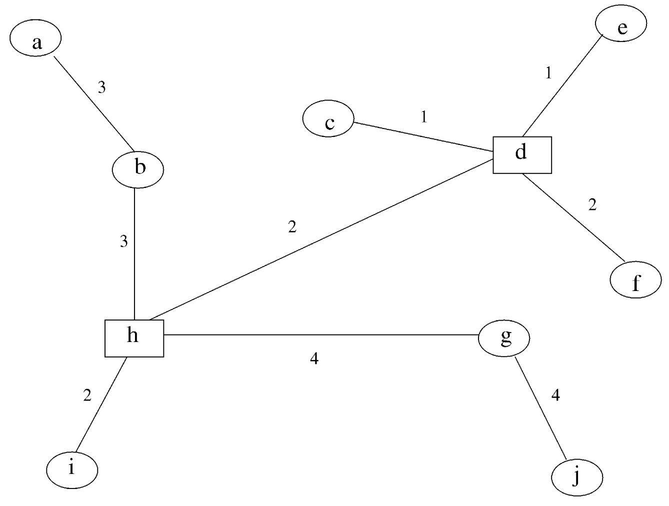

Fig. 2. A spanning lighttree with wavelength conversions at nodes d and h.

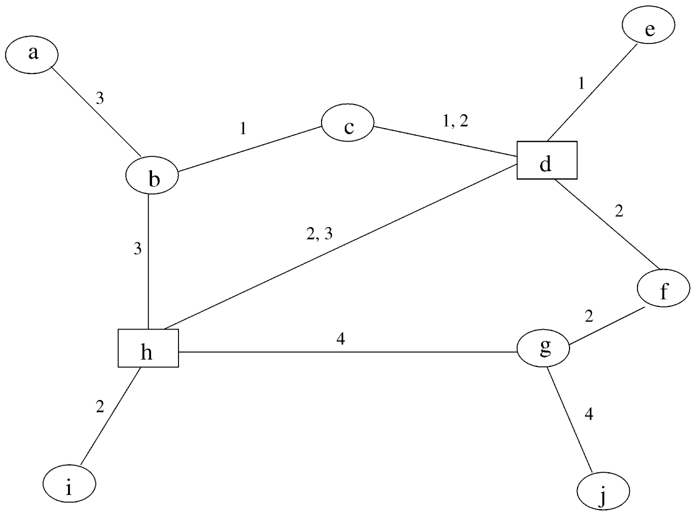

Fig. 3. $\{1, 2, 3, 4\}$ is a color-covering since the edges in these colors form a connected graph on V.

$$\begin{aligned} vc(h) &= c((b,h)) \cup c((d,h)) \cup c((g,h)) \cup c((i,h)) \\ &= \{3\} \cup \{2,3\} \cup \{4,5\} \cup \{2\} = \{2,3,4,5\}. \end{aligned}$$

A set of colors is called a *color-covering* if all edges in those colors form a connected graph on V and therefore contain a spanning tree of G. In Fig. 1, $\{1, 2, 3, 4\}$ is a color-covering since the edges with color 1, 2, 3, and 4 form a connected graph on V, as shown in Fig. 3.

A subset of vertices is called a *vertex color-covering* if all colors of those vertices form a color-covering. In Fig. 1, $\{d, h\}$ is a vertex color-covering since the colors of these two vertices are 1, 2, 3, 4, and 5 and $\{1, 2, 3, 4, 5\}$ is a color-covering of G.

A subset of vertices is said to be *color-connected* if connecting every pair of vertices that share at least one common color gives a connected graph. Given a subset of vertices $U \subseteq V$, we can construct a graph $A(U)$ in the following way to test whether U is color-connected. The vertex set of $A(U)$ equals U. For any two vertices $u_1, u_2 \in A(U)$, add an edge between them if $vc(u_1) \cap vc(u_2)$ is not empty (u_1 and u_2 share at least one common color). Now, $A(U)$ is connected if and only if U is color-connected. For example, in Fig. 1, $\{a, b, d\}$ is color-connected since $vc(a) = \{3\}$, $vc(b) = \{1, 3\}$, $vc(d) = \{1, 2, 3, 5\}$, and each pair of the three vertices share at least one color. Graph $A(\{a, b, d\})$ is a triangle and is connected. For another example, $\{a, f\}$ is not color-connected since $vc(a) = \{3\}$, $vc(f) = \{2, 5\}$, and they don't share a common color. Graph $A(\{a, f\})$ is an empty graph on a and f and is not connected.

3 Problem Formulation

In this section, we give the mathematical formulation of the Converter Placement problem. We also define two related problems: Color-Covering and Vertex Color-Covering.

We start by looking at the properties of the solutions to the Converter Placement problem. Suppose S is a solution to the Converter Placement problem for a given graph G, then S is a set of vertices in G that has the following two properties:

1. S is a vertex color-covering of G.
2. S is color-connected.

Property 1 is clear since the colors of S have to be a color-covering, otherwise, no spanning lighttree can be built using the colors of S. To see why Property 2 is true, let's look at the example network in Fig. 4a. Both $\{b, d, f\}$ and $\{a, c\}$ are vertex color-covering since both sets have colors $1, 2, 3$ and $\{1, 2, 3\}$ is a color-covering. The set $\{b, d, f\}$ is not color-connected since none of b, d, f share a common color. The set $\{a, c\}$ is color-connected since a and c share color 2. We can't build a spanning lighttree with wavelength converters placed at b, d and f, but we can build a spanning lighttree with wavelength converters placed at a and c, as shown in Fig. 4b. Since $\{b, d, f\}$ is not color-connected, the three components of color 1, color 2, and color 3 can't be joined together via $\{b, d, f\}$. On the other hand, the three components can be joined together by $\{a, c\}$ since $\{a, c\}$ is color-connected.

Now, we give the mathematical formulation of the Converter Placement problem.

Converter Placement (CP): Given a connected graph $G = (V, E)$ and a mapping c from E to a set of colors such that, for every color x, all edges in x form a connected subgraph, compute a minimum cardinality color-connected vertex color-covering of G.

In the problem formulation, we require that, for every color, all edges in the color form a connected subgraph. This may not be true in the given graph. But, we can easily modify the graph such that the condition is satisfied. Suppose, for a color x, all the edges with this color form k components $P_1, P_2, \cdots, P_k$ and $k > 1$. We can replace x by k different colors $x^1, x^2, \cdots, x^k$ and let component P_i have color x^i for $1 \leq i \leq k$. In this way, all the edges in a color form a connected subgraph. An example of this transformation is shown in Fig. 5. In Fig. 5a, color 1 has two components. We transform the graph into Fig. 5b such that one component has color 1^1 and the other component has color 1^2.

In the following, we define two problems that are closely related to Converter Placement. The problems are: Color-Covering and Vertex Color-Covering.

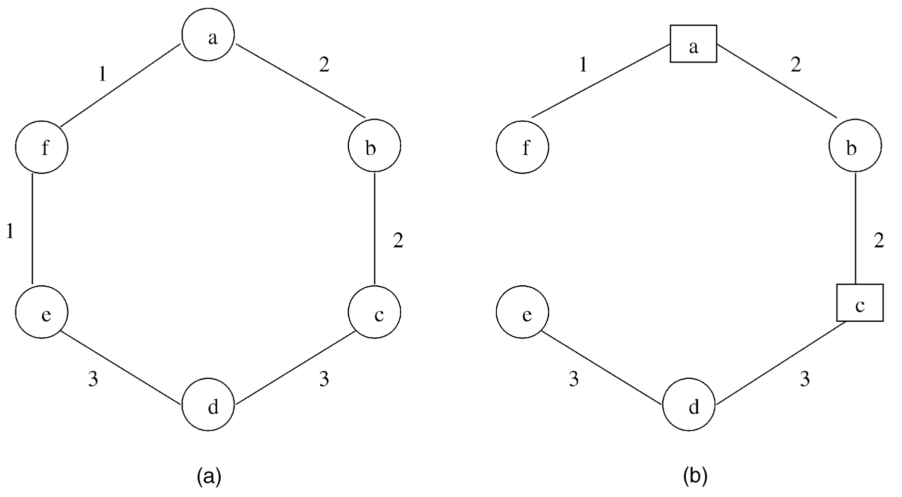

Fig. 4. (a) An example network. (b) A spanning lighttree using wavelength converters at nodes a and c.

Color-Covering (CC): Given a connected graph $G = (V, E)$ and edge color sets for each edge in G, compute a color-covering of minimum cardinality.

An optimal solution of Color-Covering is called a *minimum color-covering*. In Fig. 1, $\{1, 2, 3, 4\}$ is a minimum cover-covering.

Vertex Color-Covering (VCC): Given a connected graph $G = (V, E)$ and edge color sets for each edge in G, compute a vertex color-covering of minimum cardinality.

An optimal solution of Vertex Color-Covering is called a *minimum vertex color-covering*. In Fig. 1, $\{d, h\}$ is a minimum vertex color-covering.

In the following two sections, we study Color-Covering and Vertex Color-Covering, respectively. Both of the problems will be shown to be NP-hard and we will derive both a lower bound and an upper bound of the approximation of the two problems. Using these results, we then give some theoretical results about the hardness of the Converter Placement problem and propose a greedy algorithm to solve it in Section 6.

4 Color-Covering

In this section, we study the Color-Covering problem. We first give a lower bound for the approximation of the problem and then propose a greedy algorithm that reaches nearly-the-best performance ratio.

4.1 Lower Bound of Approximation

The following is a negative result which implies that Color-Covering is unlikely to have a polynomial-time approximation with performance ratio $\rho \ln n$ for $\rho < 1$.

Theorem 1. *Color-Covering has no polynomial-time approximation with performance ratio $\rho \ln n$ for $\rho < 1$ unless $NP \subset DTIME(n^{poly \log n})$, where n is the number of vertices in the input graph.*

Proof. The proof is based on a recent result of Raz and Safra [3] which is obtained by improving a result of Lund and Yannakakis [2]. Consider the following problem:

Set-Covering: Given a (finite) collection $\mathcal{S}$ of subsets of a set U of n elements, find a minimum cardinality subcollection of $\mathcal{S}$ such that the union of all subsets in the subcollection covers U.

Raz and Safra proved the following:

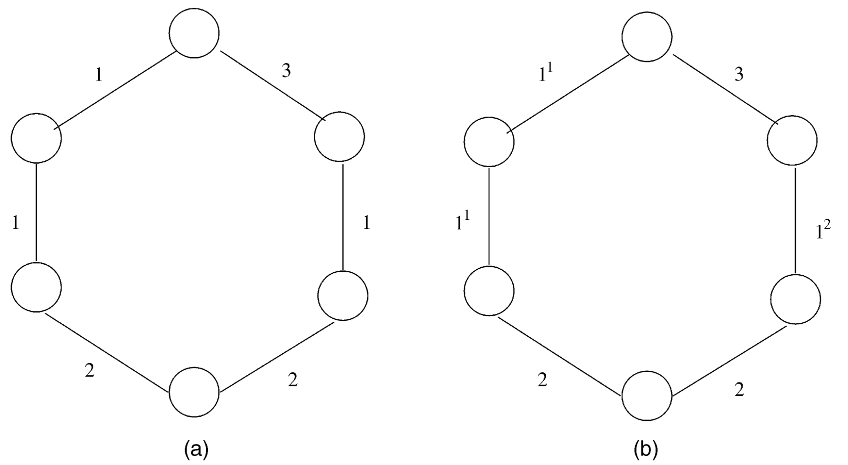

Fig. 5. (a) A graph G, where color 1 has two components. (b) After transformation, for each color, the edges form a connected subgraph.

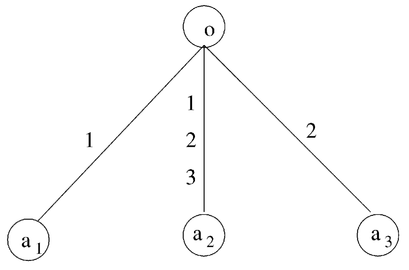

Fig. 6. Reduction from Set-Covering to Color-Covering. Edge (o, a_1) has color 1 since $a_1 \in S_1$; edge (o, a_2) has color $1, 2, 3$ since a_2 belongs to S_1, S_2, and S_3; edge (o, a_3) has color 2 since $a_2 \in S_2$. $\{S_1, S_2\}$ is a set-covering of U and $\{1, 2\}$ is a color-covering of G.

Lemma 2. *For any $0 < \rho < 1$, there is no polynomial-time approximation algorithm with performance ratio $\rho \ln n$ for the Set-Covering problem unless $NP \subset DTIME(n^{poly \log n})$.*

Now, supposing Color-Covering has a polynomial-time approximation with performance ratio $\rho \ln n$, where $\rho < 1$, we then show that Set-Covering also has a polynomial-time approximation with performance ratio $\rho \ln n$.

For any collection $\mathcal{S}$ of subsets of a set U of n elements, construct a graph G with edge-colors as follows: G has $n+1$ vertices consisting of all elements in U and a special vertex o and n edges (o, a) for each $a \in U$. Suppose $\mathcal{S} = \{S_1, S_2, \cdots, S_m\}$. Then, for each $a \in S_i$, assign a color i to edge (o, a). As an example of this construction, if we have an instance of Set-Covering: $U = \{a_1, a_2, a_3\}$, $\mathcal{S} = \{S_1, S_2, S_3\}$, $S_1 = \{a_1, a_2\}$, $S_2 = \{a_2, a_3\}$, and $S_3 = \{a_2\}$, then G is the graph shown in Fig. 6.

It's easy to verify that $\{S_{i_1}, S_{i_2}, \cdots, S_{i_k}\}$ is a set-covering of U if and only if $\{i_1, i_2, \cdots, i_k\}$ is a color-covering of G. It follows immediately that if $\{i_1, i_2, \cdots, i_k\}$ is a color-covering of G with a factor of $\rho \ln n$ from optimal, then $\{S_{i_1}, S_{i_2}, \cdots, S_{i_k}\}$ is a set-covering of U with a factor of $\rho \ln n$ from optimal. By Lemma 2, Theorem 1 is proven. □

4.2 A Greedy Algorithm

In this section, we give a polynomial-time greedy algorithm with performance ratio $\ln n + 1$ for the Color-Covering problem. Since Theorem 1 shows that Color-Covering is unlikely to have a polynomial-time approximation with performance ratio $\rho \ln n$ for $\rho < 1$, our algorithm reaches nearly-the-best possible performance ratio.

The input to our algorithm is a graph $G = (V, E)$ and a mapping c from E to a set of colors. The output is a color-covering of G. The algorithm starts with an empty set C and repeatedly chooses a color to add into C until it becomes a color-covering. In each step, we choose a color as follows: Let H be an initially empty graph on V. Let $cpnt(H)$ be the number of components in H, initially, $cpnt(H) = |V|$. Let E_i denote the set of all edges with color i. Let $r(E_i, H)$ be the number of components reduced by adding E_i to graph H. Our greedy approach is to choose the color that reduces the most number of components in H. Thus, the greedy choice maximizes $r(E_i, H)$. Whenever we choose a color i to add into C, we modify H by adding E_i to it and this will reduce $cpnt(H)$. We stop when $cpnt(H)$ is reduced to 1. At this point, the graph is connected and therefore, C is a color-covering. The pseudocode of the algorithm is given below.

Greedy Algorithm 1

```
1. input $G = (V, E)$ and a mapping c from $E$ to a set of
   colors;
2. $C \leftarrow \emptyset$;
3. $H \leftarrow$ the empty graph with vertex set $V$;
4. $E_i \leftarrow \{e \in E \mid i \in c(e)\}$;
5. while $H$ is not connected
6.     select a color $i$ that maximizes $r(E_i, H)$;
7.     $C \leftarrow C \cup \{i\}$;
8.     $E(H) \leftarrow E(H) \cup E_i$;
9. return $C$.
```

We now analyze the time complexity of Greedy Algorithm 1. Let w be the number of colors in the graph. Lines 2 and 3 take constant time. Lines 4 takes time $O(w|E|)$. The while loop in line 5 executes at most w times. In line 6, we need to take each color i not in C to compute $r(E_i, H)$ and choose a color that maximizes the reduction. The computation of $r(E_i, H)$ takes time $O(|V|)$ and there are at most w colors we can choose from, therefore, line 6 takes time $O(w|V|)$. Lines 7 and line 8 take constant time. Thus, the while loop between line 5 and line 8 takes time $O(w^2|V|)$. The running time of the entire algorithm is thus $O(w|E| + w^2|V|)$.

Theorem 3. *Greedy Algorithm 1 produces an approximation solution for Color-Covering within a factor of $\ln n + 1$ from optimal, where n is the number of vertices in the input graph.*

Proof. Suppose C is the color set obtained by Greedy Algorithm 1 and $|C| = k$. Let $1, 2, \cdots, k$ be elements of C in the order of their appearance. Denote by H_i the graph with vertex set V and edge set $E_1 \cup \cdots \cup E_i$ (H_0 is the empty graph on V). Let $s_i + 1$ be the number of components of H_i, then we have $s_k = 0$ since H_k is connected and $s_0 = n - 1$ since H_0 has n components. Denote $r_i = r(E_i, H_{i-1})$. Let C^* be an optimal solution.

For each H_{i-1}, we have

$$max_{j \in C^*} r(E_j, H_{i-1}) \geq \frac{s_{i-1}}{|C^*|}.$$

This is true since adding $\cup_{j \in C^*} E_j$ to H_{i-1} will produce a connected graph, thus reducing the number of components of H_{i-1} by s_{i-1}. So, there must be a color $j \in C^*$ such that, when E_j is added to H_{i-1}, the number of components of H_{i-1} decreases at least $\frac{s_{i-1}}{|C^*|}$.

Consider Step 6 in Greedy Algorithm 1: Since i is the color that maximizes $r(E_i, H)$, we have

$$r_i = r(E_i, H_{i-1}) \geq max_{j \in C^*} r(E_j, H_{i-1}) \geq \frac{s_{i-1}}{|C^*|}$$

and

$$s_i = s_{i-1} - r_i \leq s_{i-1} - \frac{s_{i-1}}{|C^*|} = \frac{|C^*| - 1}{|C^*|} \cdot s_{i-1}.$$

Thus,

$$1 \le s_{|C|-1} \le \left(\frac{|C^*|-1}{|C^*|}\right)^{|C|-1} \cdot s_0 = \left(\frac{|C^*|-1}{|C^*|}\right)^{|C|-1} \cdot (n-1).$$

Therefore,

$$|C| \le \log_{\frac{|C^*|}{|C^*|-1}}(n-1) + 1.$$

Dividing both sides by C^*, we have

$$\frac{|C|}{|C^*|} \le \frac{\log_{\frac{|C^*|}{|C^*|-1}}(n-1)}{|C^*|} + \frac{1}{|C^*|} \le \frac{\ln n}{|C^*| \ln \frac{|C^*|}{|C^*|-1}} + 1$$
$$= \frac{\ln n}{\ln\left(\frac{|C^*|}{|C^*|-1}\right)^{|C^*|}} + 1.$$

Since $(\frac{i}{i-1})^i \ge e$, we get

$$\frac{|C|}{|C^*|} \le \ln n + 1.$$

This proved that the solution produced by Greedy Algorithm 1 is within a factor of $\ln n + 1$ from the optimal solution. □

5 Vertex Color-Coloring

We study the Vertex Color-Covering problem in this section.

Theorem 4. *Vertex Color-Covering has no polynomial-time approximation with performance ratio $\rho \ln n$ for $\rho < 1$ unless $NP \subset DTIME(n^{poly \log n})$, where n is the number of vertices in input graph. Moreover, Vertex Color-Covering has a polynomial-time approximation with performance ratio $\ln n + 1$.*

Proof. We prove the first part of the theorem by reducing Set-Covering to Vertex Color-Covering. For an input collection $\mathcal{S}$ of subsets of a set U of m elements, construct a graph $G = (V, E)$ with edge-colors as follows: The vertex set V consists of all elements in U, all subsets in $\mathcal{S}$, and a special vertex o. The edge set E consists of (o, S) for all $S \in \mathcal{S}$ and (a, S) for all $a \in S \in \mathcal{S}$. Let $U = \{a_1, a_2, \cdots, a_m\}$. Define a mapping c from E to colors $\{1, 2, 3, \cdots, m, m+1\}$ by

$$c((o,S)) = \{m+1\} \text{ for all } S \in \mathcal{S},$$
$$c((a_i,S)) = \{i\} \text{ for all } a_i \in S \in \mathcal{S}.$$

To illustrate the reduction, let's use the same Set-Covering example as in the proof of Theorem 1. Given an instance of Set-Covering: $U = \{a_1, a_2, a_3\}$, $\mathcal{S} = \{S_1, S_2, S_3\}$, $S_1 = \{a_1, a_2\}$, $S_2 = \{a_2, a_3\}$, and $S_3 = \{a_2\}$, the reduction produces a graph G, as shown in Fig. 7.

Let V' be a vertex color-covering on $< G, c >$. Without loss of generality, we may assume $V' \subseteq \mathcal{S}$ since, otherwise, we can easily find a solution in $\mathcal{S}$ with the same or smaller cardinality by replacing each vertex not in $\mathcal{S}$ with one of its adjacent vertices in $\mathcal{S}$. For example, in Fig. 7, if o is in V', we can replace it by any one of

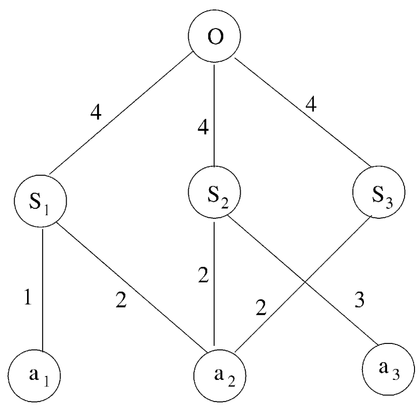

Fig. 7. Reduction from Set-Covering to Vertex Color-Covering. $\{S_1, S_2\}$ is a vertex color-covering for G and is also a set-covering for U.

S_1, S_2, S_3. If a_3 is in V', we can replace it by S_2. We claim that V' is a vertex color-covering for G if and only if V' is a set-covering for U. First, suppose V' is a vertex color-covering for G. Since $\{1, 2, 3, \cdots, m, m+1\}$ is the only color-covering for G, the colors of V' must be $1, 2, 3, \cdots, m, m+1$. Then, V' must cover $a_1, a_2, \cdots, a_m$ since a_i is connected to some vertex in V' with an edge of color i. Conversely, suppose V' is a set-covering for U, then the colors of V' must include $1, 2, \cdots, m$ plus $m+1$. Since these colors form a color-covering of G, V' is a vertex color-covering of G. Now, it's clear that an optimal solution $V^* \subseteq \mathcal{S}$ for Set-Covering is also an optimal solution for Vertex Color-Covering.

Let n be the number of vertices in G, then $n = m + |\mathcal{S}| + 1$. Suppose V' is a polynomial-time approximation for Vertex Color-Covering with performance ratio $\rho \ln n = \rho \ln(m + |\mathcal{S}| + 1)$, where $\rho < 1$. When $|\mathcal{S}| \le m$, $\rho \ln(m + |\mathcal{S}| + 1) \le \rho' \ln m$ for $\rho < \rho' < 1$ and a sufficiently large m. Therefore, V' is a polynomial-time approximation with performance ratio $\rho' \ln m$ for the Set-Covering problem in the special case $|\mathcal{S}| \le m$. From [3], we know that Lemma 2 holds in this special case, therefore Vertex Color-Covering has no polynomial-time approximation with performance ratio $\rho \ln n$ for $\rho < 1$ unless $NP \subset DTIME(n^{poly \log n})$.

Next, we prove the second part of the theorem by showing that an instance of Vertex Color-Covering can be transformed to an instance of Color-Covering and a solution to the latter problem is also a solution to the former problem.

Consider an instance $< G, c >$ of Vertex Color-Covering. Let $v_1, v_2, \cdots, v_n$ be the vertices of G. We construct an instance $< G, c' >$ of Color-Covering where c' is a mapping from E to colors $\{1, 2, \cdots, n\}$ defined by

$$c'(e) = \{i \mid e \text{ is in a color of } v_i\}.$$

Fig. 8 shows an example of the reduction from Vertex Color-Covering to Color-Covering. Fig. 8a is an instance of Vertex Color-Covering. Fig. 8b is an instance of Color-Covering constructed from Fig. 8a. In Fig. 8a, edge (v_1, v_2) has color 1, which is a color of v_1, v_2, and v_3. Therefore, in Fig. 8b, edge (v_1, v_2) has color set $\{1, 2, 3\}$. Similarly, edge (v_2, v_3) in Fig. 8b also has color set

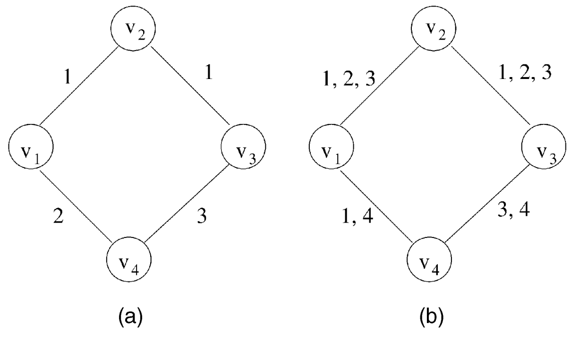

Fig. 8. (a) An instance of Vertex Color-Covering. (b) An instance of Color-Covering constructed from (a). $\{v_1\}$ is a vertex color-covering for (a) and $\{1\}$ is a color-covering for (b).

$\{1, 2, 3\}$. In Fig. 8a, edge (v_1, v_4) has color 2, which is a color of v_1 and v_4. Therefore, in Fig. 8b, edge (v_1, v_4) has color set $\{1, 4\}$. Finally, edge (v_3, v_4) has color 3, which is a color of v_3 and v_4. Therefore, in Fig. 8b, edge (v_3, v_4) has color set $\{3, 4\}$.

Now, we show that $\{v_{i_1}, v_{i_2}, \cdots, v_{i_k}\}$ is a Vertex Color-Covering on input $< G, c >$ if and only if $\{i_1, i_2, \cdots, i_k\}$ is a Color-Covering on input $< G, c' >$. First, suppose $\{v_{i_1}, v_{i_2}, \cdots, v_{i_k}\}$ is a Vertex Color-Covering on input $< G, c >$. Let C be the color set of these vertices, then there is a set of edges F in G such that F contains a spanning tree of G and each edge in F has a color in C. By the definition of c', on input $< G, c' >$, each edge in F has a color i, where $i \in \{i_1, i_2, \cdots, i_k\}$. Thus, $\{i_1, i_2, \cdots, i_k\}$ is a Color-Covering on input $< G, c' >$. Conversely, suppose $\{i_1, i_2, \cdots, i_k\}$ is a Color-Covering on input $< G, c' >$. Then, there is a set of edges F in G such that F contains a spanning tree of G and each edge in F has a color in the colors of vertices $v_{i_1}, v_{i_2}, \cdots, v_{i_k}$. Therefore, $\{v_{i_1}, v_{i_2}, \cdots, v_{i_k}\}$ is a Vertex Color-Covering on input $< G, c >$.

By Theorem 3, Color-Covering has a polynomial-time approximation with performance ratio $\ln n + 1$, therefore, Vertex Color-Covering also has a polynomial-time approximation with performance ratio $\ln n + 1$; the second half of the theorem is proven. □

6 Converter Placement

Using the results of Color-Covering and Vertex Color-Covering presented in the previous two sections, we study the hardness of the Converter Placement problem in this section and give an approximation algorithm for solving it.

6.1 Lower and Upper Bound of Approximation

Theorem 5. *Converter Placement has no polynomial-time approximation with performance ratio $\rho \ln n$ for $\rho < 1/2$ unless $NP \subset DTIME(n^{poly \log n})$, where n is the number of vertices in the input graph. Moreover, Converter Placement has a polynomial-time approximation with performance ratio $2(\ln n + 1)$.*

Proof. Let S be a solution to Vertex Color-Covering. If S is color-connected, then it is also a solution to Converter Placement. Otherwise, we can modify S to become color-connected and therefore be a solution to Converter Placement as follows: We call a subset $P \subseteq S$ a *color-connected component* if P is the vertex set of a component in graph $A(S)$. Recall that graph $A(U)$ is defined in Section 2 to test the color-connectedness of a set U. Let k be the size of S, then S has at most k color-connected components. Notice that we can add a vertex $x \in V - S$ to S such that it connects at least two color-connected components and thus reduce the number of color-connected components by at least 1. This can be done since, if we can't find such a vertex, then the graph is not connected, contradicting the fact that it is connected.

As an example, let's consider the graph in Fig. 1. $\{b, g\}$ is a vertex color-covering but is not color-connected. It has two color-connected components $\{b\}$ and $\{g\}$. We can add vertex h to connect them since h shares color 3 with b and shares colors 2, 4, 5 with g. Now, $\{b, g, h\}$ is color-connected and therefore is a solution to Converter Placement.

At most $k - 1$ vertices are needed to connect all the color-connected components in S. Therefore, there is a solution of at most $2k - 1$ vertices for Converter Placement.

Let A_{vcc} be the size of an approximation solution to Vertex Color-Covering and A_{cp} be the size of an approximation solution to Converter Placement. Let O_{vcc} be the size of an optimal solution to Vertex Color Covering and O_{cp} be the size of an optimal solution to Converter Placement. By the above argument, we have $A_{cp} \leq 2A_{vcc}$ and $O_{cp} \leq 2O_{vcc}$. Also, it's clear that $O_{cp} \geq O_{vcc}$ and $A_{cp} \geq A_{vcc}$. Thus, $\frac{A_{cp}}{O_{cp}} \geq \frac{A_{vcc}}{2O_{vcc}}$ and, therefore, $\frac{A_{vcc}}{O_{vcc}} \leq 2 * \frac{A_{cp}}{O_{cp}}$. If Converter Placement has an approximation with performance ratio $\rho \ln n$, where $\rho < 1/2$, then Vertex Color-Covering has an an approximation with performance ratio $\rho \ln n$, where $\rho < 1$. By Theorem 4, the first half of the theorem is proven.

To prove the second half of the theorem, note that $\frac{A_{cp}}{O_{cp}} \leq 2\frac{A_{vcc}}{O_{vcc}}$. By Theorem 4, Vertex Color-Covering has a polynomial-time approximation with performance ratio $\ln n + 1$, therefore, Converter Placement has a polynomial-time approximation with performance ratio $2(\ln n + 1)$. □

6.2 A Greedy Algorithm

We now give a greedy approximation algorithm for Converter Placement that achieves the approximation performance ratio $2(\ln n + 1)$. The input to our algorithm is a graph $G = (V, E)$ and a mapping c from E to a set of colors. The output is a color-connected vertex color-covering of G. The algorithm first finds a vertex color-covering of G as follows: Letting C be an initially empty set, we repeatedly choose a vertex to add into C until C becomes a vertex color-covering. In each step, we choose a vertex as follows: Let H be an initially empty graph on V. Let $cpnt(H)$ be the number of components of H, initially, $cpnt(H) = |V|$. Let E_i denote the set of all edges with color i. Let $r(E_i, H)$ be the number of components reduced by adding E_i to graph H. When we choose a vertex x to add into C, we compute $\cup_{i \in vc(x)} E_i$, which are all the edges in the vertex color set of x.

TABLE 1
Performance of Greedy Algorithm 2 on 50 Randomly Generated Graphs

Number of Vertices	Number of Optimal Solution	Number of Nonoptimal Solution	Average Performance Ratio
10	2	3	1.200
15	4	1	1.050
20	6	1	1.071
25	1	6	1.297
30	3	5	1.191
35	3	3	1.222
40	2	3	1.133
50	1	1	1.375
100	1	1	1.167
200	1	0	1.000
300	1	0	1.000
500	1	0	1.000

Adding these edges to graph H will reduce $cpnt(H)$. At each step, our greedy strategy is to pick a vertex that reduces the most number of components in H, i.e., we pick a vertex x that maximizes $r(\cup_{i\in vc(x)}E_i, H)$. Our goal is to reduce $cpnt(H)$ to 1. At this point, the graph is connected and, therefore, C is a vertex color-covering. After a vertex color-covering C is found, we check if it's color-connected. If yes, then a solution is found. Otherwise, we modify C by keep adding vertices to it until C becomes color-connected. The vertex we choose to add at each step is a vertex that reduces the most number of color-connected components in C. The pseudocode of the algorithm is given below.

Greedy Algorithm 2

```
1.  input $G = (V, E)$ and a mapping $c$ from $E$ to a set of
    colors;
2.  $C \leftarrow \emptyset$;
3.  $H \leftarrow$ the empty graph with vertex set $V$;
4.  $E_i \leftarrow \{e \in E \mid i \in c(e)\}$;
5.  while$H$ is not connected
6.      select a vertex $x$ that maximizes $r(\cup_{i\in vc(x)}E_i, H)$;
7.      $C \leftarrow C \cup \{x\}$;
8.      $E(H) \leftarrow E(H) \cup \cup_{i\in vc(x)}E_i$;
9.  while$C$ is not color-connected
10.     select a vertex $x$ such that $C \cup \{x\}$ has the least
        number of color-connected components;
11.     $C \leftarrow C \cup \{x\}$;
12. return $C$.
```

We now analyze the time complexity of Greedy Algorithm 2. Let w be the number of colors in the graph. Lines 2 and 3 take constant time. Line 4 takes time $O(w|E|)$. The while loop in line 5 executes at most $|V|$ times. In line 6, we need to take each vertex x not in C to compute $r(\cup_{i\in vc(x)}E_i, H)$ and choose one vertex that maximizes the reduction. The computation of $r(\cup_{i\in vc(x)}E_i, H)$ takes time $O(w|V|)$ and there are at most $|V|$ vertices we can choose from, therefore, line 6 takes time $O(w|V|^2)$. Line 7 takes constant time and line 8 takes time $O(w)$. Thus, the while loop between line 5 and line 8 takes time $O(|V|(w|V|^2 + w))$, which is $O(w|V|^3)$. For the loop between line 9 and line 11, it executes at most $|V|$ times and, in each loop, line 10 dominates the running time. In line 10, we need to take each vertex x not in C to compute the number of color-connected components in $C \cup \{x\}$ and choose the vertex that produces the least number of color-connected components. The computation of the number of color-connected components in $C \cup \{x\}$ takes time $O(w|V|)$ and there are at most $|V|$ vertices we can choose from, therefore, line 10 takes time $O(w|V|^2)$. Thus, the while loop between line 9 and line 11 takes time $O(w|V|^3)$. The running time of the entire algorithm is thus $O(w|E| + w|V|^3)$. Furthermore, since $|E|$ is $O(|V|^2)$, the running time is $O(w|V|^3)$.

7 Numerical Results

To evaluate the performance of Greedy Algorithm 2, we implemented the algorithm in C++. We also implemented a brute force (BF) algorithm to compute the optimal solution so that we can obtain the performance ratio of our algorithm. Suppose the input network has n nodes, BF computes an optimal solution as follows: It generates subsets of $\{1, 2, \cdots, n\}$ from size 1 to size n. Whenever a subset is generated, it is tested to see whether the subset is a solution to Converter Placement. If it's a solution, then the algorithm stops. Otherwise, it continues to generate the next subset and checks again. BF guarantees to produce the optimal solution, but the running time is exponential to n and therefore is not useful in practice.

We run both Greedy Algorithm 2 and BF on 50 randomly generated graphs. The number of vertices ranges from 10 to 500 and the number of colors ranges from four to 64. The results showed that, out of the 50 test cases, Greedy Algorithm 2 produced optimal solution in 26 cases. A summary of the test results is shown in Table 1. Column 1 is the number of vertices in the test cases. Column 2 is the number of test cases that produce an optimal solution. Column 3 is the number of test cases that produce a nonoptimal solution. The last column is the average performance ratio of the test cases. The average performance ratio over the 50 test cases is 1.169. Since it is fairly close to 1, we conclude that Greedy Algorithm 2 produces good approximation solutions to the Converter Placement problem.

8 Conclusion

In this paper, we studied the problem of finding the minimum set of network nodes to place converters to support broadcast in WDM optical networks. We presented a mathematical formulation of the Converter Placement problem. Theoretical lower bound and upper bound are given for the approximation of the problem by studying two related problems, namely Color-Covering and Vertex Color-Covering. Two approximation algorithms are given. Greedy Algorithm 1 solves the Color-Covering problem and Greedy Algorithm 2 solves the Converter Placement Problem. We evaluated the performance of Greedy Algorithm 2 and our results showed that it achieves the optimal solution in 52 percent of the time and the average performance ratio is 1.169.

ACKNOWLEDGMENTS

The authors wish to thank Professor Manuel Blum for helpful discussion and comments. The authors are grateful for the valuable comments given by the anonymous reviewers. Research supported in part by the US National Science Foundation under grant CCR-9530306 and 973 Fundamental Research Project of China.

REFERENCES

[1] C.A. Brackett, "Dense Wavelength Division Multiplexing Networks: Principles and Applications," *IEEE J. Selected Areas in Comm.,* vol. 8, pp. 948-964, Aug. 1990.

[2] C. Lund and M. Yannakakis, "On the Hardness of Approximating Minimization Problems," *J. ACM,* vol. 41, no. 5, pp. 960-981, Sept. 1994.

[3] R. Raz and S. Safra, "A Sub-Constant Error-Probability Low-Degree Test, and a Sub-Constant Error-Probability PCP Characterization of NP," *Proc. 29th Symp. Theory of Computing (STOC),* pp. 475-484, May 1997.

[4] M. Kovacevic and A. Acampora, "Benefits of Wavelength Translation in All-Optical Clear-Channel Networks," *IEEE J. Selected Areas in Comm.,* vol. 14, pp. 868-880, June 1996.

[5] R. Ramaswami and K.J. Sivarajan, "Routing and Wavelength Assignment in All-Optical Networks," *IEEE/ACM Trans. Networking,* vol. 3, no. 5, pp. 489-500, Oct. 1995.

[6] B. Ramamurthy and B. Mukherjee, "Wavelength Conversion in WDM Networking," *IEEE J. Selected Areas in Comm.,* vol. 16, no. 7, pp. 1061-1073, Sept. 1998.

[7] T.E. Stern and K. Bala, *Multiwavelength Optical Networks: A Layered Approach.* Reading, Mass.: Addison-Wesley, 1999.

[8] L.H. Sahasrabuddhe and B. Mukherjee, "Light-Trees: Optical Multicasting for Improved Performance in Wavelength-Routed Networks," *IEEE Comm. Magazine,* vol. 37, no. 2, pp. 67-73, Feb. 1999.

[9] D. Li, X. Du, X. Hu, L. Ruan, and X. Jia, "Minimizing Number of Wavelengths in Multicast Routing Trees in WDM Networks," *Networks,* vol. 35, no. 4, pp. 260-265, July 2000.

[10] X. Zhang, J. Wei, and C. Qiao, "Constrained Multicast Routing in WDM Networks with Sparse Light Splitting," *Proc. INFOCOM 2000,* pp. 1781-1790, Mar. 2000.

[11] R.K. Pankaj, "Wavelength Requirements for Multicasting in All-Optical Networks," *IEEE/ACM Trans. Networking,* vol. 7, no. 3, pp. 414-424, June 1999.

[12] N. Sreenath, K. Satheesh, G. Mohan, and C. Murthy, "Virtual Source Based Multicast Routing in WDM Optical Networks," *Proc. Int'l Conf. Networking (ICON) 2000,* pp. 385-389, Sept. 2000.

[13] G. Xiao and Y.W. Leung, "Algorithms for Allocating Wavelength Converters in All-Optical Network," *IEEE/ACM Trans. Networking,* vol. 7, no. 4, pp. 545-557, Aug. 1999.

[14] S. Thiagarajan and A.K. Somani, "An Efficient Algorithm for Optimal Wavelength Converter Placement on Wavelength-Routed Networks with Arbitrary Topologies," *Proc. INFOCOM '99,* pp. 916-923, Mar. 1999.

[15] S. Subramaniam, M. Azizoglu, and A.K. Somani, "On Optimal Converter Placement in Wavelength-Routed Networks," *IEEE/ACM Trans. Networking,* vol. 7, no. 5, pp. 754-766, Oct. 1999.

Lu Ruan received the BE degree in computer science from Tsinghua University, Beijing, China, in 1996. She received the MS degree in computer science from the University of Minnesota-Twin Cities in 1999. She is currently a PhD candidate in the Department of Computer Science and Engineering, University of Minnesota-Twin Cities and will receive her PhD degree in May 2001. Her research interests include computer networks and analysis and design of optimization algorithms.

Dingzhu Du received the MS degree in 1982 from Institute of Applied Mathematics, Chinese Academy of Sciences, and the PhD degree in 1985 from the University of California at Santa Barbara. He worked at the Mathematical Sciences Research Institute, Berkeley, California, in 1985-1986, at the Massachusetts Institute of Technology in 1986-1987, and at Princeton University in 1990-1991. Currently, he is a professor in the Department of Computer Science, University of Minnesota and also a research professor at the Institute of Applied Mathematics, Chinese Academy of Sciences. His research interests include combinatorial optimization, communication networks, and theory of computation.

Xiaodong Hu received the BS degree in applied mathematics from Qinghua University, Beijing, China, in 1985 and the PhD degree in operations research and cybernetics from the Institute of Applied Mathematics, Chinese Academy of Sciences, in 1989. He was a postdoctororal fellow at the Rutgers Center for Operations Research, Rutgers University, in 1990-1991, a visiting associate professor in the Graduate School of Information Science, Japan Advanced Institute of Science and Technology, in 1993-1994, and a research fellow in the Computer Science Department, City University of Hong Kong in 1998-2000. Currently, he is a research professor at the Institute of Applied Mathematics, Chinese Academy of Sciences. His research interests include combinatorial optimization and computer and communication networks. He is a member of the IEEE Computer Society.

Xiaohua Jia received the DSc degree in information science from the University of Tokyo, Japan, in 1991. He is currently an associate professor in the Department of Computer Science at the City University of Hong Kong. His research interests include distributed systems, computer networks, real-time communications, and Internet computing. He is a senior member of the IEEE.

Deying Li received the MS degree in mathematics in 1988 from Central China Normal University, China. She is currently a PhD candidate in the Department of Computer Science, City University of Hong Kong. Her main research interests include optical networks and design of optimization algorithms.

Zheng Sun received the BS and MS degrees in mathematics from Shandong University, China. He is currently a member of the teaching staff at Shandong University. His research interests include combinatorics, graph theory, and algorithms. He has published several papers in those areas.