Routing and Wavelength Assignment in WDM Mesh Networks

Dirceu Cavendish
NEC Laboratories America
10080 North Wolfe Road
Cupertino, CA 95014, USA
e-mail: dirceu@nec-labs.com

Aleksandar Kolarov
NEC Laboratories America
4 Independence Way
Princeton, NJ 08540, USA
e-mail: kolarov@nec-labs.com

Bhaskar Sengupta1
ExxonMobil Research and Engineering
1545 Route 22 East
Annandale, NJ 08801, USA
e-mail: Bhaskar.Sengupta@exxonmobil.com

Abstract—In this paper, we study routing and wavelength assignment (RWA) tasks in WDM mesh networks with wavelength conversion capabilities. For each lightpath request, we solve the routing and wavelength assignment sub-problems as a single problem, rather than separating them into two separate problems. We use four different optimization criteria: (1) minimize the number of wavelength conversions (2) minimize the number of wavelengths used (3) minimize the hop count and (4) minimize the use of scarce resources (trunking). Each of these problems is solved by a Dijkstra type of algorithm, executed sequentially on one request at a time. We also develop an integer linear programming (ILP) model with the objective to minimize the number of wavelength conversions. The optimal ILP solution can only be obtained for small networks. We conduct a comprehensive simulation study of the performance of the algorithms for different network topologies and traffic scenarios.

I. INTRODUCTION AND RELATED WORK

Routing and wavelength assignment (RWA) is the procedure by which routing paths are determined and wavelengths are assigned to connections to be provisioned by an optical transport network (OTN). The RWA should be done so as to conserve OTN resources as much as possible, since this may result in a lower blocking probability when additional connections are requested. Designing the system so as to directly minimize blocking probability is quite a difficult problem. So, in this paper, we focus on four different criteria for optimization as follows: 1) Minimizing the number of wavelength conversions used; 2) Minimizing the number of wavelengths used; 3) Minimizing the total hop count used by all lightpaths; 4) Minimizing the use of “scarce resources”. We note that the methods developed in this paper can be easily extended to optimize objective functions which combine two or more of the criteria given above.

The RWA problem takes two flavors: off-line and on-line [1]. In the off-line case, all connection requests are known in advance, thus a routing decision can be made based on the complete knowledge of the traffic to be served by the network. In the on-line case, a connection request must be routed and wavelengths assigned independently of other connections, which either have already been assigned or will be assigned in the future. Finding a globally optimal solution for the off-line problem is NP-hard. So one is forced to use algorithms which are sub-optimal but are computationally efficient for the off-line problem.

Various aspects of the routing and wavelength assignment problem in mesh networks have been addressed in the recent literature. The integer linear programming (ILP) approach to address the RWA problem is considered in [2]–[5]. Among the RWA problems with no wavelength conversion capability, a quite comprehensive RWA problem formulation is presented in [6], [7]. In [7] the authors developed an integer linear programming model, which for a given set of lightpath requests, determines the routes and assigns wavelengths to the lightpaths so as to minimize the number of ports needed. They described the properties of a node instead of a link as in other ILP formulations for RWA, but they did not consider optical cross-connects with wavelength conversion capability. The authors of [8] presented an ILP formulation of the RWA problem with wavelength conversion, attesting that only very small instances of the problem can be solved optimally. They also introduced a detailed modeling of optical cross-connects (OXC), including wavelength conversion and multiplexing. They presented a heuristic based on $K$ alternate shortest paths and define a wavelength assignment strategy seeking to minimize the number of conversions needed. In [9], the authors introduced an optical node architecture that deals with multigranularity flows, including fiber, wavelength and waveband. Graph transformation techniques were also used by [10] to solve RWA problems, via minimization of a path cost function which involves link costs with a traversal together with a wavelength conversion components.

The work presented in this paper differs from previous work in several ways. First, we consider WDM mesh networks with a specified limitation on wavelength conversion capabilities (number of conversions). Second, we formulate problems with different objectives for optimization, whereas most of the previous work has focused on a single objective (e.g., minimizing the number of wavelengths used). Third, we develop an ILP model with the objective to minimize the number of wavelength conversions. The ILP models with other objective functions are not considered due to lack of space for their presentations. The formulation of the ILP model is unique in the future. Finding a globally optimal solution for the off-line problem is NP-hard. So one is forced to use algorithms which are sub-optimal but are computationally efficient for the off-line problem.

1This work was performed while the author was with Yeshiva University, New York.
since it describes a detailed model of optical cross-connects with wavelength conversion capability. Fourth, for each of the objectives outlined earlier, we develop a sequential algorithm (by sequential we mean one request considered singly and sequentially). This allows us to use the algorithms for both off-line and on-line RWA problems. Each algorithm, based on the Dijkstra algorithm, solves the routing and wavelength assignment sub-problems as a single problem, rather than separating them into a routing and a wavelength assignment sub-problems (e.g., in approaches using alternate paths). Although our basic approach is to use a graph transformation and apply a Dijkstra approach is to use a graph transformation and apply a Dijkstra type algorithm, which may resemble previous work [10], we evaluate various cost functions from a network performance point of view.

The paper is organized as follows. In section II, we state the problem. In section III we present an ILP model with the objective to minimize the number of wavelength conversions. In sections IV-VII, we describe the proposed algorithms for solving mesh RWA problems. In section VIII, we evaluate the performance of the proposed algorithms. Conclusions are drawn in section IX.

II. PROBLEM STATEMENT

A WDM mesh network of \( N \) nodes is modeled as a directed graph \( G(V, E) \), where \( |V| = N \), and \( |E| = J \). Nodes are labeled by \( n \), where \( 0 \leq n \leq N - 1 \). A link \((n, m)\) connects node \( n \) to node \( m \). The presence of link \((n, m)\) in \( E \) means that communication can take place from node \( n \) to node \( m \). We represent the graph \( G \) by means of adjacency lists. So, for all \( j \in V \), we define \( A(j) \) to be a subset of \( V \), containing the nodes adjacent to node \( j \), i.e., \( n \in A(j) \) if and only if \((n,j) \in E\).

A wavelength set \( \Lambda = \{0, 1, \ldots, K - 1\} \) of size \( |\Lambda| = K \) represents the number of wavelengths (colors) available in the WDM network. For each node \( j \) of the mesh network, we specify a number \( \kappa_j \), which denotes the maximum number of wavelengths which can be converted at node \( j \). An OEO switch fabric at node \( j \) has \( \kappa_j \) input-output ports and tunable lasers, and therefore, \( \kappa_j \) typically a power of 2. In view of the cost of the OEO switch, we expect \( \kappa_j \) to be much smaller than \( K \).

A request set \( R = \{(s_0, d_0), \ldots, (s_{|R|-1}, d_{|R|-1})\} \) is defined as a set of pairs \( s_i \) and \( d_i \), of size \( |R| \), representing the source and destination nodes of the connections to be provisioned. We require \( s_i \neq d_i \). From an element \((s_i, d_i)\) (also called a connection) of the request set, we must construct a lightpath \( L_i \), which is defined as a sequence of vertices \( L_i = (s_i, I_1(1), I_2(2), \ldots, I_{\ell_i}(\ell_i), d_i) \), where \( \{I_j(j); j = 1, \ldots, \ell_i\} \) represent intermediate nodes in the path from \( s_i \) to \( d_i \) for request \( i \). Note that it is possible for \( \ell_i \) to be zero. The coloring of lightpath \( L \) is defined as the assignment of wavelength labels to all links of \( L \). If two consecutive links in a lightpath are colored with different colors, say colors \( r \) and \( s \), the node \( j \) attached to these two links must be capable of performing this wavelength conversion. This means that the value of \( \kappa_j \) must be greater than or equal to the number of lightpaths that require wavelength conversion at node \( j \).

The problem now is to route and color each and every element of the request set so as to optimize one of the criteria mentioned earlier. The global optimization problem (when all requests are considered together as in the off-line version of the problem) is NP-hard. In the next section we formulate an integer linear programming (ILP) model with the objective to minimize the number of wavelength conversions. The ILP models with different objective functions (criteria 2, 3, and 4) can similarly be formulated, but they are omitted in this paper due to space limitations.

III. INTEGER LINEAR PROGRAMMING MODEL

In this section we first describe the OXC model. The nodes in the optical network are hybrid optical devices each consisting of a transparent optical (OOO) and a wavelength opaque (OEO) switch. This architecture is referred to as “hybrid cross-connect”. Fig. 1 shows the architecture of a hybrid OXC.

Each hybrid node has several ports that are used as terminal points for both external and local add/drop links. Input links are terminated at input ports, and output links are originated from output ports. Local input (add) and output (drop) ports are the part of the OEO switch. We assume that the number of add ports and the number of drop ports are not limited and equal to the number of wavelength requests to be added and dropped at a given OXC, respectively. The number of input and output ports of OEO switch that is used for wavelength conversions when provisioning connections through the OXC is limited to \( \kappa \) ports. The number of input and output ports of OOO switch is limited.

As Fig. 1 shows, an optical connection can be provisioned through the hybrid OXC by one of the following ways:

**OOO**: From a non-local input port to a non-local output port through the OOO switch (no wavelength conversion);

**OEO**: From a non-local input port to a non-local output port through the OEO switch (wavelength conversion). The maximum number of wavelength conversions is limited to \( \kappa \);

**Drop**: From a non-local input port to a local drop port through the OEO switch (no wavelength conversion);

**Add**: From a local add port to a non-local output port through the OEO switch (no wavelength conversion).
A. Notations

\( \mathcal{V} \): Set of network nodes;

\( N \): Number of network nodes;

\((n, m)\): Link between node \( n \) and node \( m \);

\( \mathcal{A}(n) \): Set of adjacent nodes to node \( n \);

\( \mathcal{P}_i^n \): Set of input ports at node \( n \) where incoming links \((m, n), m \in \mathcal{A}(n)\) are terminated;

\( \mathcal{P}_o^n \): Set of output ports at node \( n \) where outgoing links \((n, m), m \in \mathcal{A}(n)\) are originated;

\( \mathcal{P}_{i,m}^n \): Input port at node \( n \) where an incoming link from node \( m \) is terminated; thus, \( \mathcal{P}_i^n = \bigcup_{m} \mathcal{P}_{i,m}^n \);

\( \mathcal{P}_{o,m}^n \): Output port at node \( n \) where an outgoing link to node \( m \) is originated; thus, \( \mathcal{P}_o^n = \bigcup_{m} \mathcal{P}_{o,m}^n \);

\( \mathcal{L}_i^n \): Set of local add ports at node \( n \);

\( \mathcal{L}_o^n \): Set of local drop ports at node \( n \);

\( \mathcal{P}\mathcal{L}_i^n \): \( \mathcal{P}_i^n \cup \mathcal{L}_i^n \). Set of all input ports (local and non-local) at node \( n \);

\( \mathcal{P}\mathcal{L}_o^n \): \( \mathcal{P}_o^n \cup \mathcal{L}_o^n \). Set of all output ports (local and non-local) at node \( n \);

\( K \): Number of wavelengths available in a WDM network;

\( \Lambda \): Wavelength set \( \Lambda = \{0, 1, \ldots, K - 1\} \);

\( k, l, p \): Wavelengths from set \( \Lambda \);

\( \kappa_n \): Maximum number of wavelengths which can be converted at node \( n \);

\( \eta_n \): Number of wavelengths converted at node \( n \);

\( T \): Number of node pairs having non-zero traffic demand;

\( c_t \): Number of connection requests to be provisioned between the node pair \((s_t, d_t)\), where \( s_t \) and \( d_t \) represent the source and destination node, respectively. Note that \( t = (0, \ldots, T - 1) \);

\( \mathcal{R}_t \): Set of connection requests \( \mathcal{R}_t = \{r_{t}^{0}, \ldots, r_{t}^{c_t-1}\} \) of size \( c_t \), to be provisioned between the node pair \((s_t, d_t)\);

\( \mathcal{R} \): Set of all connection requests \( \mathcal{R} = \bigcup_{t} \mathcal{R}_t \), of size \( |\mathcal{R}| = \sum_{t=0}^{T-1} c_t \), to be provisioned in a given network topology. Note that \( \mathcal{R} = \{r_{t}^{j} : 0 \leq t < T, 0 \leq j < c_t\} \).

B. ILP Variables

The following variables describe the properties of the hybrid OXC node.

\( W_{r_{t}^{j},i,o}^{n,k,l} \): 1 if at node \( n \) a connection request \( r_{t}^{j} \) \((r_{t}^{j} \in \mathcal{R})\) uses wavelength \( k \) at input port \( i \) \((i \in \mathcal{P}\mathcal{L}_i^n)\) and wavelength \( l \) at output port \( o \) \((o \in \mathcal{P}\mathcal{L}_o^n)\); otherwise, it is 0;

The traffic at a node can be drop traffic, through traffic or add traffic. The variable \( W_{r_{t}^{j},i,o}^{n,k,l} \) represents drop traffic when \( i \in \mathcal{P}_i^n, o \in \mathcal{L}_o^n \) (item "Drop"); through traffic when \( i \in \mathcal{P}_i^n, o \in \mathcal{P}_o^n \) (item "OOF" or "OEO") or add traffic when \( i \in \mathcal{L}_i^n, o \in \mathcal{P}_o^n \) (item "Add").

C. Minimizing the Number of Wavelength Conversions

Let \( \eta_n \) be the number of wavelengths converted at node \( n \) (see item "OEO"). Our first objective is to minimize the total number of OEO conversions in a WDM network. A path using the smallest number of hops is favored when there is a tie between multiple paths with the same number of wavelength conversions.

\[
\min \sum_{n \in \mathcal{V}} \eta_n + \epsilon \omega_n,
\]

where

\[
\eta_n = \sum_{r_{t}^{j} \in \mathcal{R}_n, i \in \mathcal{P}_i^n, o \in \mathcal{P}_o^n, k, l \in \Lambda} W_{r_{t}^{j},i,o}^{n,k,l},
\]

\[
\omega_n = \sum_{r_{t}^{j} \in \mathcal{R}_n, i \in \mathcal{P}_i^n, o \in \mathcal{P}_o^n, k, l \in \Lambda} W_{r_{t}^{j},i,o}^{n,k,l},
\]

\[
\epsilon < (KN)^{-1}.
\]

In (1), \( \eta_n \) represents the number of OEO conversions at node \( n \), and \( \omega_n \) represents the number of active wavelengths on the input ports of the same node. The small constant \( \epsilon \) is used for breaking ties between the paths with the same number of OEO conversions. Note that the sum \( \sum_{n \in \mathcal{V}} \omega_n \) represents the total number of hops of provisioned requests.

D. Constraints

In the formulation of the RWA ILP model, we first specify constraints on traffic flows. We assume that a connection request \( r_{t}^{j} \), namely the \( j \)-th connection request between the node pair \((s_t, d_t)\), \((0 \leq t < T, 0 \leq j < c_t)\), is given. The constraints for the request \( r_{t}^{j} \) in case when \( r_{t}^{j} \) passes through node \( n \) are defined as follows.

\[
\sum_{i \in \mathcal{L}_i^n, o \in \mathcal{P}_o^n, k, l \in \Lambda} W_{r_{t}^{j},i,o}^{n,k,l} = 0 \quad n \neq s_t, n \neq d_t,
\]

\[
\sum_{i \in \mathcal{P}_i^n, o \in \mathcal{L}_o^n, k, l \in \Lambda} W_{r_{t}^{j},i,o}^{n,k,l} = 0 \quad n \neq s_t, n \neq d_t.
\]

If node \( n \) is either the source or destination node, then the constraints are given as follows.

\[
\sum_{i \in \mathcal{P}_i^n, o \in \mathcal{P}_o^n, k, l \in \Lambda} W_{r_{t}^{j},i,o}^{n,k,l} = 0 \quad n = s_t,
\]

\[
\sum_{i \in \mathcal{P}_i^n, o \in \mathcal{P}_o^n, k, l \in \Lambda} W_{r_{t}^{j},i,o}^{n,k,l} = 0 \quad n = d_t.
\]

We also require \( s_t \neq d_t \), what can be specified as follows.

\[
\sum_{r_{t}^{j} \in \mathcal{R}_n, i \in \mathcal{P}_i^n, o \in \mathcal{P}_o^n, k, l \in \Lambda} W_{r_{t}^{j},i,o}^{n,k,l} = 0.
\]
Next, we specify the wavelength capacity constraint on both input and output ports of node \( n \).

\[
\sum_{r_i^t \in R, r_o \in P_{L}^n, l \in \Gamma} W_{r_i^t,i,o}^{n,k,l} \leq 1 \quad i \in P_i^n, k \in \Lambda, \quad (12)
\]

\[
\sum_{r_i^t \in R, r_o \in P_{L}^n, l \in \Gamma} W_{r_i^t,i,o}^{n,k,l} \leq 1 \quad o \in P_o^n, l \in \Lambda. \quad (13)
\]

Let us now consider a link \((m,n)\) connecting nodes \( m \) and \( n \). Then, for any wavelength \( l \) \((l \in \Lambda)\) and request \( r_i^t \) \((r_i^t \in R)\) the wavelength continuity constraint on link \((m,n)\) is defined as follows.

\[
\sum_{i \in P_i^m, k \in \Lambda} W_{i,i}^{m,k,l} - \sum_{o \in P_o^m, p \in A} W_{i,p}^{n,l,p} = 0
\]

\[
r_i^t \in R, m, n \in V, l \in \Lambda. \quad (14)
\]

The constraint on the number of OEO conversions at node \( n \) \((\eta_n)\) is defined as follows.

\[
\sum_{r_i^t \in R, i \in P_i^m, o \in P_o^m, k,l \in \Lambda, l \neq k} W_{r_i^t,i,o}^{n,k,l} \leq \kappa_n. \quad (15)
\]

The ILP model just presented can be used to solve RWA problems for networks whose lightpaths must be provisioned for a set of pre-defined requests. This is a common scenario on layered networks, where other circuit and packet technologies run on top of lightpaths. In this case, the ILP formulation above provides an efficient traffic engineering solution for avoiding costly wavelength conversion hardware. Moreover, large RWA problems may be solvable since they can be solved “off-line”.

In case “on-line” solutions of the RWA problem is needed, the ILP formulation developed is not applicable firstly because knowledge of the entire request set is not feasible, and secondly because even if that was the case, via reconfiguration of all lightpaths plus a new request, there might not be enough time to solve a large RWA problem every time a new request arrives. For these cases, we need to develop efficient heuristics. In the next four sections, we examine a class of algorithms with polynomial complexity, which are sequential. Sequential optimization problem is likely to occur in many optical networks where connection requests come in one at a time.

IV. MINIMIZING THE NUMBER OF WAVELENGTH CONVERSIONS

We first describe an algorithm to minimize the number of wavelength conversions used. If there are multiple ways in which this can be achieved, then the algorithm minimizes the length of the lightpath. Readers interested in the details of the Dijkstra algorithm are referred to [11].

Algorithm begin:

Step 1: First order the request set in some order. One way of ordering them would be to find the shortest paths for each element of the set and order them by the length of the shortest path (increasing or decreasing). If ties remain, then order them further by the index of the source node. If ties still remain, then order them further by the index of the destination node. At this time, if ties remain, it implies that there are multiple requests with the same source and destination nodes, which are indistinguishable. For all \( n, m \in V \) and \( k \in \Lambda \) define \( \alpha_{nm}(k) = 1 \), if wavelength \( k \) is available for use on that link, and \( 0 \) otherwise. Usually, \( \alpha_{nm}(k) = 1 \) for \( k = 0, \cdots, K - 1 \) and \((n,m) \in E\), unless for some reason, certain wavelengths are not available on some links (e.g., out of service). Also, if \((n,m) \notin E\), then \( \alpha_{nm}(k) = 0 \). Let \( \beta_{k} = 1 \) for \( k = 0, \cdots, K - 1 \). Let \( H \) denote the total number of hops in all paths selected so far and initialize \( H \) to \( 0 \). Let \( \epsilon \) denote a small quantity satisfying \( \epsilon < (KN)^{-1} \). For all \( j \in V \), let \( A(j) \) be a subset of \( V \) containing nodes adjacent to node \( j \), i.e., \( n \in A(j) \) if and only if \((n,j) \in E\). Let \( \rho_{j} = \kappa_{j} \) for all \( j \in V \). Select the first request of the ordered request set.

Step 2: Renumber the nodes of the network so that the source node of the request under consideration is numbered 0 and the destination node is numbered \( N - 1 \). Since this renumbering is a permutation of the indices of the original network, permute the quantities \( \alpha_{nj}(k), \beta_{j} \) and \( A(j) \) in a similar manner, for all \( j, n \in V \). Let the set \( V^* = \{1, \cdots, N-1\} \), i.e., \( V^* = V - \{0\} \). Let \( W \) denote the set of all tuples of the form \((n,k)\) where \( n \in V^* \) and \( k \in \Lambda \). Let \( M \) denote a subset of \( W \) and initialize \( M \) to the empty set. For all \( j \in V^* \) and \( k \in \Lambda \), let \( \Gamma(j,k) = 0 \) and \( \Pi(j,k) = 0 \).

Step 3: For all \( n \in V^*, j \in A(n) \) and \( k, l \in \Lambda \), let

\[
d(n,k;j,l) = \begin{cases} \epsilon & \text{if } k = l \text{ and } \alpha_{nj}(l) = 1, \\ 1 + \epsilon & \text{if } k \neq l \text{ and } \alpha_{nj}(l) = 1 \text{ and } \rho_{n} \geq 1, \\ \infty & \text{otherwise}. \end{cases}
\]

For all \( j \in V^* \) and \( k \in \Lambda \), let

\[
f(j,k) = \begin{cases} \epsilon & \text{if } \alpha_{0j}(k) = 1, \\ \infty & \text{otherwise}. \end{cases}
\]

Step 4: Let

\[
g = \min_{(j,k) \in W - M} f(j,k). \quad (18)
\]

Let \((\mu, \nu)\) be the value of \((j,k)\) which minimizes (18). Resolve ties by first taking the lowest index of the \( j \) variable and if ties still remain, then the lowest index of the \( k \) variable. Let \( M \leftarrow M \cup \{\mu, \nu\} \). If \( \mu = N - 1 \) then go to step 5; otherwise continue. For all \( j \in A(\mu) \) and \( k \in \Lambda \):

\[
\text{if } \{f(j, k) > f(\mu, \nu) + d(\mu, \nu, j, k)\} \quad (19)
\]

\[
\text{then } \{f(j, k) = f(\mu, \nu) + d(\mu, \nu, j, k)\} \quad (20)
\]

\[
\Pi(j,k) = \mu \quad (21)
\]

\[
\Gamma(j,k) = \nu. \quad (22)
\]

Repeat step 4.

Step 5: Let \( m = 1, \xi_{m} = \mu \) and \( \phi_{m} = \nu \). If \( f(\xi_{m}, \phi_{m}) = \infty \), stop. There is no path available for this request. Select the next request (if any left) from the request set and go to step

IEEE Communications Society
Globecom 2004
1019
0-7803-8794-5/04/$20.00 © 2004 IEEE
2. Otherwise, continue.

\[
\text{while } (\xi_m \neq 0) \\
\{ m \leftarrow m + 1 \\
\phi_m = \Gamma(\xi_{m-1}, \phi_{m-1}) \\
\xi_m = \Pi(\xi_{m-1}, \phi_{m-1}) \}\]

**Step 6:** Let \(\alpha_{i_1 \ldots i_j}(\phi_j) = 0\) for \(j = 1, \ldots, m - 1\). Let \(\beta_{\phi_j} = 0\) for \(j = 1, \ldots, m - 1\). Let \(\rho_{\phi_j} \leftarrow \Delta(\phi_{j-1}, \phi_j)\) for \(j = 2, \ldots, m - 1\), where \(\Delta(x, y) = 1\) if \(x \neq y\) and 0 otherwise. Let \(H = H + m - 1\). If requests are left, then pick a new request from the request set and go to step 2. If no requests are left, then stop.

**End of Algorithm**

For the request under consideration, the optimal lightpath consists of \(m\) nodes and equals \((\xi_{m}, \xi_{m-1}, \ldots, \xi_1)\), where \(\xi_m = 0\) and \(\xi_1 = N - 1\). The wavelength used on the link \((i_1, i_2)\) is \(\phi_i\) for \(i = 1, \ldots, m - 1\). The number of wavelength conversions needed by the request under consideration is given by \(\sum_{i=2}^{m-1} \Delta(\phi_{i-1}, \phi_i)\). The variable \(\beta_k\) is 1 if wavelength \(k\) has been used by any request provisioned so far and it is 0 otherwise. Hence the number of wavelengths used by all the requests processed so far is given by \(K - \sum_{k \in A} \beta_k\). The hop count for the request under consideration is given by \(m - 1\). The hop count of all requests processed so far is given by \(H\). The value of \(\rho_j\) for \(j = 0, \ldots, N - 1\) denotes the wavelength conversion capability still left at node \(j\), after the processing of all requests so far. This means that no wavelength conversion capability is left at node \(j\) if \(\rho_j\) has dropped to a value of 0. The value of \(\sum_{k \in A} \alpha_{nm}(k)\) represents the total number of wavelengths left for use on the link \((n, m)\) for all \(n, m \in \mathcal{V}\), after taking into account the wavelengths used by all requests processed so far.

**V. Minimizing the Number of Wavelengths Used**

This method minimizes the number of wavelengths used, one request at a time. If ties remain, it picks a path with the fewest number of wavelength conversions. The method of solving the problem is identical to the one presented in section IV, except step 3, which should be changed to:

**Step 3:** For all \(n \in \mathcal{V}^*\), \(j \in \mathcal{A}(n)\) and \(k, l \in \Lambda\), let

\[
d(n, k; j, l) = \begin{cases} 
N^l \beta_l & \text{if } k = l \text{ and } \alpha_{nj}(l) = 1 \\
N^l \beta_l + \epsilon & \text{if } k \neq l, \alpha_{nj}(l) = 1 \text{ and } \rho_n \geq 1, \\
\infty & \text{otherwise.}
\end{cases}
\]

For all \(j \in \mathcal{V}^*\) and \(k, l \in \Lambda\), let

\[
f(j, k) = \begin{cases} 
N^k \beta_k & \text{if } \alpha_{0j}(k) = 1, \\
\infty & \text{otherwise.}
\end{cases}
\]

**VI. Minimizing the Hop Count**

This method minimizes the number of hops used, one request at a time. If there are multiple ways in which this can be achieved, then it minimizes the number of wavelength conversions. The method of solving the problem is identical to the one presented in section IV, except step 3, which should be changed to:

**Step 3:** For all \(n \in \mathcal{V}^*\), \(j \in \mathcal{A}(n)\) and \(k, l \in \Lambda\), let

\[
d(n, k; j, l) = \begin{cases} 
1 & \text{if } k = l \text{ and } \alpha_{nj}(l) = 1 \\
1 + \epsilon & \text{if } k \neq l, \alpha_{nj}(l) = 1 \text{ and } \rho_n \geq 1, \\
\infty & \text{otherwise.}
\end{cases}
\]

For all \(j \in \mathcal{V}^*\) and \(k, l \in \Lambda\), let

\[
f(j, k) = \begin{cases} 
1 & \text{if } \alpha_{0j}(k) = 1, \\
\infty & \text{otherwise.}
\end{cases}
\]

**VII. Minimizing the Use of Scarce Resources**

This method conserves the use of critical resources, whether it is wavelengths available on a link or wavelength conversion capability at a node. We make use of two thresholds, \(C\) and \(T\). If on any particular link, \(C\) or less wavelengths are currently available for use, then the algorithm does not permit the use of any more wavelengths on that link as long as other alternative paths are available. Note that this restriction is somewhat different from the one imposed in circuit-switched networks, where the last \(C\) circuits (or trunks) on a link can be used for one hop calls only. Similarly, if at a particular node, the wavelength conversion capability has fallen to \(T\) or less, this node does not perform any more wavelength conversions as long as other paths are available. When this criterion produces a tie between two or more paths, the algorithm resolves the tie by minimizing the number of hops used. If there are multiple ways in which this can be achieved, then it minimizes the number of wavelength conversions. The method of solving the problem is identical to the one presented in section IV, except step 3, which should be changed to:

**Step 3:** For all \(n, j \in \mathcal{V}^*\), let \(\theta(n, j) = 1\) if \(\sum_{k=0}^{K-1} \alpha_{nj}(k) \leq C\); otherwise \(\theta(n, j) = 0\). For all \(n \in \mathcal{V}^*, j \in \mathcal{A}(n)\) and \(k, l \in \Lambda\), let

\[
d(n, k; j, l) = \begin{cases} 
1 & \text{if } \theta(0, j) = 0 \text{ and } \alpha_{0j}(k) = 1, \\
1 + N & \text{if } \theta(0, j) = 1 \text{ and } \alpha_{0j}(k) = 1, \\
1 + \epsilon & \text{if } k \neq l, \alpha_{nj}(l) = 1, \theta(n, j) = 1 \text{ and } \rho_n > T, \\
1 + \epsilon & \text{if } k \neq l, \alpha_{nj}(l) = 1, \theta(n, j) = 1 \text{ and } \rho_n < T, \\
\infty & \text{otherwise.}
\end{cases}
\]

**VIII. Performance Analysis**

In this section, we report on simulation results about the four routing and wavelength assignment (RWA) algorithms for mesh WDM networks introduced earlier. The four RWA strategies used are: MinConv - minimizes the number of wavelength conversions; MinWav - minimizes the number of wavelengths used; MinHop - minimizes the number of hops used; Trunking - minimizes the use of critical resources,
whether it is wavelengths available on a link or wavelength conversion capability at a node. We first study how close to the best RWA solution our MinConv algorithm performs, by comparing its performance against the solution of ILP formulation of the RWA problem, given in section III. Given the complexity of the ILP formulation, only small RWA problems can be handled. Large problems are dealt with in a subsequent subsection, where our heuristics are compared to one another.

A. Optimum RWA Solution and Our Heuristic

We use two network topologies to compare the performance of the MinConv algorithm with the optimum solution of the ILP problem, obtained by the CPLEX software tool. The first topology is the ring network with $N = 6$ nodes and $J = 6$ bidirectional links. The second topology is the NSF backbone optical network depicted in Fig. 2, which has $N = 13$ nodes and $J = 16$ bidirectional links. Each link has a capacity of 4 wavelengths in each direction ($K = 4$).

For each network topology, a traffic matrix is randomly generated based on the uniform distribution. The number of lightpaths requested by any source-destination node pair is in the range $0 - 4$. The total number of lightpath requests in the ring network is 31 and in the NSF network is 54. We define the following two metrics: the total number of wavelength conversions in the network, denoted by $\Omega$, and the hop count of all requests processed, denoted by $H$. The results are given in Table I.

<table>
<thead>
<tr>
<th>Topology</th>
<th>ILP Model</th>
<th>MinConv</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ring</td>
<td>$\Omega_1$</td>
<td>$H_1$</td>
</tr>
<tr>
<td>NSF</td>
<td>$\Omega_2$</td>
<td>$H_2$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table I</th>
</tr>
</thead>
<tbody>
<tr>
<td>Results for the optimum and MinConv algorithms</td>
</tr>
</tbody>
</table>

From Table I, we see that the hop count of all requests processed under the MinConv algorithm is identical to that of the optimum solution (ILP Model). As expected, the total number of wavelength conversions under the MinConv algorithm is always larger than the optimum one. The MinConv algorithm sequentially processes requests, as opposed to the ILP model, which considers all requests at once. Hence, the wavelength assignment under the MinConv algorithm is less efficient than the optimum one, what causes some unnecessary wavelength conversions.

B. Heuristics Performance for Large RWA Problems

We use several performance metrics for comparison of the RWA algorithms. The first one is the network blocking probability, denoted as $B$, which expresses the ratio of the number of rejected lightpath requests to the total number of requests. The second performance metric, denoted by $\gamma$, is the average number of hops per request, averaged over all provisioned requests. The third parameter, denoted as $\eta$, is the average number of wavelength conversions per request, averaged over all provisioned requests.

Due to space limitations we only present a set of results for one randomly generated mesh network. We use a method by Waxman [12] for generating random mesh topologies. The network topology consists of $N = 32$ nodes and $J = 79$ bidirectional links. Each link has a capacity of 16 wavelengths in each direction ($K = 16$). We simulated three traffic types, based on the uniform, hub and dual-hub traffic distributions. In case of the uniform distribution, the source node and the destination node of a lightpath request are selected randomly. In case of the hub distribution, one randomly selected node is first designated as the hub node. Each request uses this hub node either as its source or destination (with equal probability). The other node of the request is selected randomly. In case of the hub distribution, two nodes are selected randomly and are designated as hub nodes. Each request uses one of these two hub nodes (with equal probability) as its source or destination node (again with equal probability). The other node of the request is selected randomly. We simulated different traffic load scenarios by varying the number of requests, with the following choices: $|R| = 64, 128, 256, 512$ and 1024.

The parameters $\kappa$, $C$, and $T$ were set at 8 (for all $j$), 2, and 1 respectively. Each point in any graph is an average of 5 simulations. The results for $B$, $\gamma$ and $\eta$ are shown in Figs. 3 through 5, respectively. From the results in Fig. 3 - 5, we draw the following conclusions:

- For all network and traffic topologies and under all algorithms, the blocking probability increases when the...
traffic load increases. Blocking probability is largest for the hub traffic distribution and smallest for the uniform traffic distribution. The easiest request set to route and color comes from the uniform traffic distribution.

- In terms of blocking probability, the MinWav algorithm has the worst performance of all four algorithms. The reason for this is that minimizing the number of wavelengths used increases the average path length (see the values of $\eta$ in Fig. 4). Thus, longer paths are chosen so as to spare the use of additional wavelengths, which ends up taking more network resources and increasing blocking probability. The MinConv and MinHop algorithms show almost the same performance. The blocking probability for the Trunking algorithm is pretty much the same as that of the MinConv and MinHop algorithms.

- As expected, the MinConv strategy achieves the minimum number of wavelength conversions (see $\eta$ in Fig. 5), whereas the MinHop strategy minimizes the number of hops (see $\gamma$ in Fig. 4). The average number of hops for the Trunking algorithm is slightly worse than that of MinConv and MinHop algorithms.

Although minimization of the number of wavelengths used (MinWav) has been a popular strategy with many researchers, our empirical evidence shows that this strategy is easily the worst, both in terms of blocking probability and in terms of the number of hops. The other three strategies are almost identical for blocking probability and exhibit some differences in number of hops.

**IX. CONCLUSIONS**

In this paper, we proposed four different optimization criteria to solve the RWA problem in mesh WDM networks: minimization of hop count; minimization of the number of wavelengths used; minimization of the wavelength conversions; trunking. For each of the four proposed algorithms, we developed a method of finding the optimal sequential solution for the on-line RWA problem, i.e., one request at a time, by using a Dijkstra type algorithm. We also developed an integer linear programming (ILP) model with the objective to minimize the number of wavelength conversions. Our unique ILP model can handle a detailed model of hybrid OXC nodes. The solution of the ILP model is a globally optimal solution for the off-line RWA problem, although it can only be obtained for relatively small networks, due to its exponential complexity. We studied the performance of the algorithms for a comprehensive set of network topologies and traffic scenarios. Regardless of the network topology and traffic scenario, we have drawn the following conclusions. The RWA method which minimizes the hop count (MinHop) and the method which minimizes the number of wavelength conversions (MinConv) are the best and achieve comparable blocking probabilities and network costs. The algorithm that minimizes the use of critical resources (Trunking) achieves almost the same blocking probability as those two algorithms but at a slightly higher network cost. We have also found that minimizing the number of wavelengths used is the worst strategy in terms of network performance (blocking probability) and network cost, which is the most popular RWA strategy.

**REFERENCES**