An Efficient Algorithm for Virtual-Wavelength-Path Routing Minimizing Average Number of Hops

Harsha V. Madhyastha
Department of Computer Science and Engineering
Indian Institute of Technology, Madras
Chennai, India
email : harsha@peacock.iitm.ernet.in

N. Balakrishnan
Supercomputer Education and Research Center
Indian Institute of Science
Bangalore, India
email : balki@serc.iisc.ernet.in

***Abstract*—In this paper, we present a novel heuristic for routing and wavelength assignment in Virtual-Wavelength-Path (VWP) routed WDM optical networks. We are the first to take up the approach of both minimizing the network cost as well as maximizing the resource utilization. Our algorithm not only minimizes the number of wavelengths required for supporting the given traffic demand on any given topology, but also aims to minimize the mean hop length of all the lightpaths which in turn maximizes the resource utilization. The algorithm initially assigns the minimum hop path to each route and then performs efficient rerouting to reduce the number of wavelengths required while also trying to minimize the average hop length. To further reduce the network cost, we also propose a wavelength assignment procedure for VWP routed networks which minimizes the number of wavelength converters required. Our algorithm has been tested on various topologies for different types of traffic demands and has been found to give solutions much better than previous standards for this problem.**

***Index Terms*—WDM optical network, Virtual-Wavelength-Path routed network, Routing and wavelength assignment, Wavelength conversion, Network cost, Resource utilization**

I. Introduction

With the rapid growth of the Internet and the ever-increasing demand for voice and video transmission, Wavelength Division Multiplexed (WDM) optical networks have assumed prime importance. By allowing several channels to be routed on the same fiber on different wavelengths, the capacity of each link is increased tremendously. However, this also calls for more efficient planning before provisioning lightpaths. The problem of assigning routes and wavelengths to lightpaths, called the Routing and Wavelength Assignment (RWA) problem, has been studied widely in literature [1] [2]. As provisioning of an extra wavelength involves considerable increase in network cost, the objective is to minimize the number of wavelengths required, called the Network Wavelength Requirement (NWR). The main constraint on this problem is the fact that the cross-connects at each node are assumed not to have any wavelength conversion capability which implies that the same wavelength is assigned to a lightpath on all the links along which it is routed. Such networks are said to be Wavelength-Path (WP) routed. This constraint can be eased in Virtual-Wavelength-Path (VWP) routed networks which was introduced in [3]. Here, all cross-connects are assumed to have full wavelength conversion capability, *i.e.*, any incoming lightpath can be assigned to any wavelength on the output side.

With the removal of the wavelength-continuity constraint, the problems of routing and wavelength assignment become independent. Now, the NWR becomes equal to the the Link Wavelength Requirement (LWR) of the maximum loaded link since wavelengths to a lightpath can be assigned independently on each link through which it passes. Hence, the problem of minimizing NWR reduces to that of minimizing the maximum LWR. Yet, the routing algorithm itself is NP-Hard and there can exist no deterministic algorithm which gets to the optimal solution always. Though wavelengths can be assigned at random after the routing phase, it is prudent to efficiently allocate the wavelengths so as to minimize the number of converters required because wavelength converters also add to the overall network cost.

Out of the literature already existing in this area, the most efficient RWA algorithm for VWP routed networks has been that proposed by Nagatsu, Hamazumi and Sato [4]. This heuristic initially follows priority-based routing, where the priority is the product of the minimum number of hops between the source and the destination and the number of channels yet to be routed between them. The route assigned is the one with minimum sum of link weights where the weightage of each link is the number of channels already routed through it. In the next phase, rerouting is done to reduce the maximum LWR. A genetic algorithm based on ant-colony optimization was proposed for this problem by Varela and Sinclair [5]. Though it performed admirably well, it did not match up to the standards of [4]. Another scheme was presented, specifically for the COST 239 European Optical Network, by Tan and Sinclair [6]. This problem was also tackled using an ILP formulation of the problem by Wauters and Demeester [7]. The efficacy of using ILP formulations for solving this problem is very less as the time and space complexity involved are huge and it becomes impractical to use these for large networks with dense traffic.

In our work, we adopt a new approach towards solving this problem of RWA in VWP routed networks. Apart from minimizing the NWR in order to reduce the network cost, we also take up the objective of maximizing resource utilization. One of the standard metrics for resource utilization is average weighted hop count. This is defined as the average number of physical hops traversed by one unit of traffic. In our problem setting, the smallest traffic unit is the amount of traffic that can be carried on a single lightpath. Hence, average weighted hop count is equivalent to the average number of physical hops taken up by a lightpath. Based on this point of view, we propose a heuristic algorithm for routing lightpaths to minimize both NWR and the average hop length of a lightpath. Though the problem of wavelength assignment is disjoint from that of routing in VWP routed networks, it is necessary to assign wavelengths so as to minimize the number of wavelength conversions because wavelength converters add to the overall cost of the network. We present a wavelength assignment algorithm for this purpose which aims to minimize the number of wavelength converters required. Therefore, our routing algorithm when used in combination with the wavelength assignment procedure we propose will achieve the objective of minimizing network cost as well as maximizing resource utilization.

The remaining part of this paper is organized as follows. In Section II, we explain in detail the heuristic algorithm we propose for routing. We then outline the procedure for wavelength assignment in Section III. The results of the simulations we conducted to compare the performance of our heuristic with that of [4] are presented in Section IV. Finally, we conclude and lay down some directions for future work in this area in Section V.

II. Routing Algorithm

As explained in the previous section, our approach to finding the optimal solution for the Routing and Wavelength Assignment (RWA) problem in Virtual-Wavelength-Path (VWP) routed networks tries to not only minimize the number of wavelengths required (called the Network Wavelength Requirement - NWR) but also aims to minimize the average hop length. The input required for this problem is the physical topology of the network under consideration and the number of lightpaths required to be established between each pair of nodes in this network. Initially, each lightpath is assigned to the route with minimum number of physical hops between source and destination (which can be determined using Dijkstra's algorithm). Later, rerouting is performed to reduce the maximum Link Wavelength Requirement (LWR), which is also the NWR, while also trying to minimize the average hop length. Assume that the links are stored in decreasing order of LWR in a list called **LINKS** and associated with each link **L** is a list called **ROUTES** which stores the lightpaths passing through link **L**, in the form **(source, destination)**, in increasing order of number of hops. The rerouting procedure is as follows :

1) Consider the first link **L** in the list **LINKS**.
2) Consider the first route **(S, D)** in the list **ROUTES** for the link **L**.
3) Let the route presently assigned from **S** to **D** be **S** $\rightarrow N_{i_0} \rightarrow N_{i_1} \rightarrow \ldots \rightarrow N_{i_h} \rightarrow$ **D** where $N_{i_x} \rightarrow N_{i_{x+1}}$ forms the link **L**.
4) Set k to x.
5) Add all neighbours of node N_{i_k} except $N_{i_{k-1}}$ and $N_{i_{k+1}}$ (either of them may not exist if k = 0 or h) to the list **NHBRS**.
6) If the list **NHBRS** is empty, skip to step 9.
7) Consider the first node **M** in the list **NHBRS**.
8) If the route between N_{i_k} and $N_{i_{x+1}}$ is rerouted as $N_{i_k} \rightarrow M \rightarrow (path\ P) \rightarrow N_{i_{x+1}}$, (where path **P** is the path with minimum hops from **M** to $N_{i_{x+1}}$), will the load on link $N_{i_k} \rightarrow M$ as well as on each of the links along the path **P** be lesser than the load on link **L**?
    a) If yes, change the route from **S** to **D** as **S** $\rightarrow N_{i_0} \rightarrow \ldots \rightarrow N_{i_k} \rightarrow M \rightarrow (path\ P) \rightarrow N_{i_{x+1}} \rightarrow \ldots \rightarrow$ **D**, update load on each link and go back to step 1.
    b) If no, remove node **M** from the list **NHBRS** and go back to step 6.
9) if k > 0, decrement k and go back to step 5.
10) Set k to x + 1.
11) Add all neighbours of node N_{i_k} except $N_{i_{k-1}}$ and $N_{i_{k+1}}$ (either of them may not exist if k = 0 or h) to the list **NHBRS**.
12) If the list **NHBRS** is empty, skip to step 15.
13) Consider the first node **M** in the list **NHBRS**.
14) If the route between N_{i_x} and N_{i_k} is rerouted as $N_{i_x} \rightarrow (path\ P) \rightarrow M \rightarrow N_{i_k}$, (where path **P** is the path with minimum hops from N_{i_x} to **M**), will the load on link $M \rightarrow N_{i_k}$ as well as on each of the links along the path **P** be lesser than the load on link **L**?
    a) If yes, change the route from **S** to **D** as **S** $\rightarrow N_{i_0} \rightarrow \ldots \rightarrow N_{i_x} \rightarrow (path\ P) \rightarrow M \rightarrow N_{i_k} \rightarrow \ldots \rightarrow$ **D**, update load on each link and go back to step 1.
    b) If no, remove node **M** from the list **NHBRS** and go back to step 12.
15) if k $< $ h, increment k and go back to step 11.
16) Consider the next route **(S, D)** in the list **ROUTES** for the link **L** and go back to step 3. If there is no route left to be considered, skip to next step.

17) Consider the next link **L** in the list **LINKS** and go back to step 2. If there is no link left to be considered, then the algorithm terminates.

The essence of the above given algorithm can be summarized as follows. After the initial routing stage, wherein we assign minimum hop path to each lightpath, we try to minimize the load on the link(s) with maximum LWR (say link **L** between nodes **a** and **b**) by rerouting some lightpath which passes through it. We consider the lightpaths in increasing order of number of hops (taken up by the currently assigned route) because the scope available for rerouting of shorter lightpaths is more (higher number of links are free implies higher degrees of freedom). For this, we partition the set of nodes, through which the lightpath (which is currently under consideration, say **R**) passes, into 2 subsets - one containing all the nodes occurring before the link **L**, *i.e.*, lesser number of hops away from **a** than **b** (along the route), say set **A** and the other containing all the nodes occurring after the link, *i.e.*, lesser number of hops away from **b** than **a** (along the route), say set **B**.

Now, we consider the nodes in set **A** in increasing order of number of hops away from **a**. For each node (say node **N**), we enumerate all its neighbours, other than the ones adjacent to it on the current route. These neighbours are considered in random order and for each neighbour (say node **M**), we check whether, if the portion of the lightpath from **N** to **b** is rerouted through the link between **N** and **M** followed by the minimum hop path from **M** to **b**, the load on each of the links through which this rerouted portion passes is lesser than that on link **L**. If it is, the route for the lightpath **R** is changed as follows. The route from source to **N** is retained as before, followed by the link **N** $\rightarrow$ **M**, followed by the minimum hop path from **M** to **b**, followed by the path from **b** to destination as in the original route.

If even after considering all the nodes in set **A**, no rerouting was possible, then the similar procedure is repeated with the nodes in set **B**. The only difference being that for these nodes, we try to find an alternate path passing through the minimum path from one of their neighbours to the node **a**. If the lightpath **R** could not be rerouted, we move onto the next route (with least number of hops among the remaining routes) through link **L** and try rerouting it. If all routes on link **L** have been considered, we move onto the next link (the one with maximum LWR among the remaining links). We finally stop when no route on any link can be rerouted. If at any stage rerouting was possible, we start all over again with the least hop path through the maximum loaded link.

The points to be noted in the above rerouting scheme are that first of all, rerouting ensures that the new route does not pass through the link **L** (**a** $\rightarrow$ **b**). So, every rerouting ensures that the traffic on the link under consideration is reduced. The links are considered in decreasing order of LWR, as NWR is the same as the maximum LWR and hence, reducing NWR requires rerouting of some lightpath passing through the link with maximum LWR. Also, the load on each of the links through which the

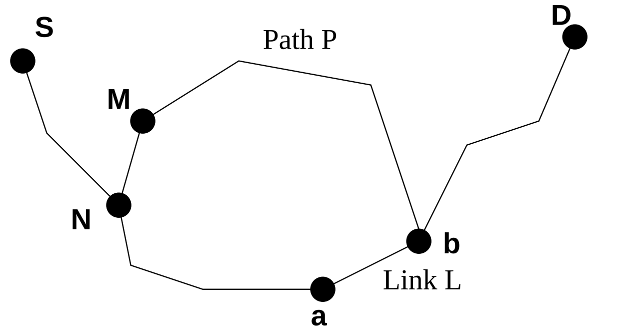

Fig. 1. Rerouting from some node in set **A** to **b**

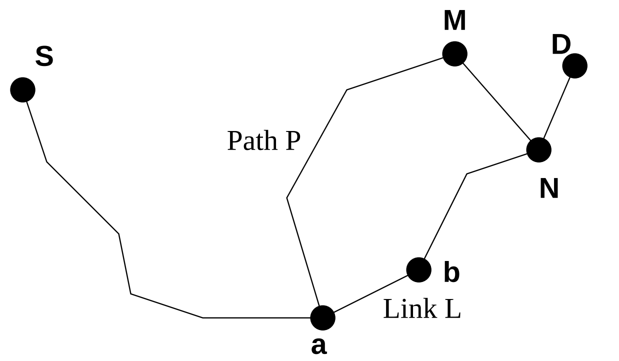

Fig. 2. Rerouting from some node in set **B** to **a**

rerouting is done is lesser than the load that was on link **L** before rerouting. This ensures that the algorithm converges to a final solution and terminates in a finite amount of time. However, the other significant point to be noted, the one which helps to minimize the average number of hops, is that whenever a lightpath is rerouted, the number of hops on the lightpath can increase by at most 2 hops. This will be made clear by Fig. 1 and Fig. 2.

In Fig. 1, since Path **P** is the minimum hop path from **M** to **b**,

$$No.\ of\ hops\ on\ Path\ \mathbf{P} <= No.\ of\ hops\ from\ \mathbf{M}\ to\ \mathbf{N} + No.\ of\ hops\ from\ \mathbf{N}\ to\ \mathbf{b}\ on\ the\ current\ route \quad (1)$$

Since there is a link between **M** and **N**, this reduces to

$$No.\ of\ hops\ on\ Path\ \mathbf{P} <= 1\ + No.\ of\ hops\ from\ \mathbf{N}\ to\ \mathbf{b}\ on\ the\ current\ route \quad (2)$$

Now, the increase in the number of hops on the lightpath is given by

$$Increase\ in\ No.\ of\ hops = No.\ of\ hops\ on\ Path\ \mathbf{P}\ + 1\ -\ No.\ of\ hops\ from\ \mathbf{N}\ to\ \mathbf{b}\ on\ the\ current\ route \quad (3)$$

Using equations 2 and 3,

$$Increase\ in\ no.\ of\ hops <= 2 \quad (4)$$

The proof for the case when rerouting is done from a node in set **B** to **a** is similar. So, in either case, the maximum increase in number of hops on the lightpath from **S** to **D** is 2. Using this constrained form of rerouting, we not only manage to minimize the maximum LWR (and hence, the NWR) but also minimize the average number of hops.

In light of the fact that the rerouting we employ ensures that the number of physical hops on the lightpath chosen for rerouting does not increase by more than 2, a further optimization can be done to our routing algorithm. In step 5 of our algorithm, we add all the neighbours of node N_{i_k} to the list **NHBRS**. This operation must be carried out such that when the nodes in the list are accessed (node **M** in step 7), they are done so in the increasing order of number of hops on the shortest path from **M** to $N_{i_{x+1}}$, *i.e.*, number of hops on path **P**. This ensures that rerouting is always performed along the shortest available path and even though the number of hops cannot increase by more than 2, this additional step further increases the probability of rerouting leading to a reduction in the number of hops. Similar considerations must also be taken into account in step 11.

III. Wavelength Assignment

After performing the routing to minimize the NWR, we need to assign wavelengths for each lightpath on each of the links through which it is routed. In order to reduce the network cost, the wavelength assignment has to be done so as to minimize the number of wavelength converters required. At each node **N**, one wavelength converter is required for each lightpath **R** which passes through it such that **R** has been assigned different wavelengths on the 2 links incident at **N** through which **R** passes. For the wavelength assignment we follow a greedy approach, which is slightly similar to the one followed in [4]. The algorithm given in [4] is meant for WP routed networks and so, the number of wavelengths required is decided by the wavelength assignment procedure. However, in our case, as we are considering VWP routed networks, the number of wavelengths required is determined by the routing procedure itself as NWR is equal to the maximum LWR. The purpose of our wavelength assignment procedure is to assign the lightpaths to the wavelengths in the range 1 to NWR, minimizing the number of wavelength converters. The wavelengths to be assigned can be assumed to be sequentially numbered from 1 to NWR. Also assume that all the lightpaths are arranged in decreasing order of number of hops in the list **ROUTES**. Let the function, assigned(**W**, **L**), take the value 1 if wavelength **W** has been assigned to some lightpath on link **L** or else 0.

1) Set **WAVE-NUM** to 1.
2) Let **R** be the first lightpath in the list **ROUTES**.
3) Is $\sum_{\mathbf{L}}$ assigned(**WAVE-NUM**, **L**) = 0 over all the links **L** through which lightpath **R** passes? If yes, assign wavelength **WAVE-NUM** to the lightpath **R** on all its links and remove lightpath **R** from the list **ROUTES**.
4) If all the lightpaths in the list **ROUTES** have already been considered, skip to next step. Else, let **R** be the next lightpath in the list **ROUTES** and go back to step 3.
5) If **WAVE-NUM** is not equal to NWR, increment **WAVE-NUM** and go back to step 2.
6) Let **R** be the first lightpath in the list **ROUTES**.
7) Let **W** be the serial number of the wavelength which minimizes $\sum_{\mathbf{L}}$ assigned(**W**, **L**) over all the links **L**, through which lightpath **R** passes and on which lightpath **R** has not yet been assigned a wavelength. If more than one **W** satisfies this property, select the least **W** among them.
8) Assign wavelength **W** to the lightpath **R** on all the links **L**, through which lightpath **R** passes and on which lightpath **R** has not yet been assigned a wavelength and where assigned(**W**, **L**) = 0.
9) If lightpath **R** has not been assigned a wavelength on all the links through which it passes, go back to step 7. Else, remove lightpath **R** from the list **ROUTES**.
10) If the list **ROUTES** is empty, then the algorithm terminates. Else, go back to step 6.

The synoptic explanation of the above algorithm is as follows. First, we take up each wavelength sequentially, trying to assign lightpaths to that wavelength in priority order (path with higher number of hops has higher priority) such that the same wavelength can be assigned to that lightpath on all the links through which it passes. When no lightpath can be assigned to a particular wavelength, we move on to the next wavelength and start assigning lightpaths to it. When all the wavelengths (from 1 to NWR) have been considered, we move onto the next phase. Here, we consider the lightpaths in decreasing order of number of hops. For each lightpath **R**, we determine the wavelength **W** which is as of now unassigned on the maximum number of links (compared to other wavelengths), through which the lightpath **R** passes and on which the lightpath **R** has not been assigned a wavelength yet. The lightpath **R** is assigned to wavelength **W** on all the links where it has not been assigned a wavelength yet and on which wavelength **W** is unassigned. We repeat the previous 2 steps until lightpath **R** has been assigned to some wavelength on each of the links through which it passes. Then, we move on to the next lightpath (the one with maximum number of hops among the remaining lightpaths) and repeat the process. Throughout this procedure, we only consider wavelengths in the range 1 to NWR.

IV. Results

To determine the optimality of our algorithm, we tested it on networks of various physical topologies with a wide variety of traffic distributions. We compared our results with that obtained by using the heuristic proposed in [4], the one currently considered to be the best for this problem. Owing to the extra constraints in our rerouting procedure, our heuristic certainly took more number of iterations in the rerouting phase to get to the final solution. As our problem setting involves static planning of the network before provisioning lightpaths, the time of

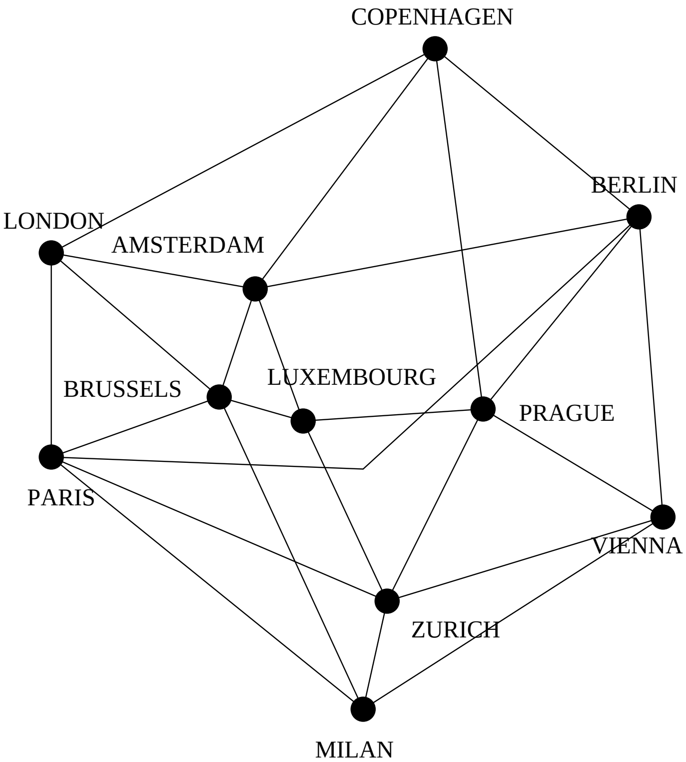

Fig. 3. Topology of European Optical Network

execution is not much of an issue. In our case, the time of execution of our heuristic was more or less comparable to that of [4], with both of them taking only a few seconds even on the real-world physical topologies and traffic demands we considered. Though the final solution, in terms of NWR, given by both heuristics was more or less the same in most cases, the average hop length was considerably lesser with our heuristic in almost all cases considered.

Here we give the results of testing on 3 standard networks along with their corresponding measured traffic demands as given in the literature. The first network considered is the Pan-European Optical Network (given in [8]), which has 19 nodes and 39 links, with the traffic demand as given in [9]. The second

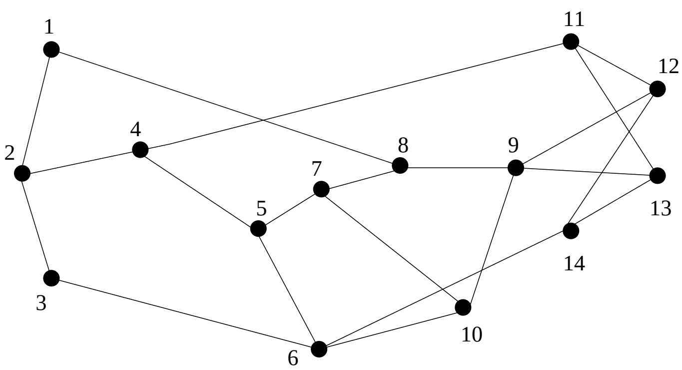

Fig. 4. Topology of NSFNET

TABLE I
RESULTS OBTAINED

S.No.	Nagatsu		Min-Hops	
	NWR	Total No. of hops	NWR	Total No. of hops
1	67	534	69	440
2	26	138	25	113
3	6	101	6	85

network considered is the NSFNET (shown in Fig. 4), which has 14 nodes and 21 links, with the measured traffic demand taken from [10]. The topology considered for the last network is formed from the 11 central nodes of the European Optical Network (shown in Fig. 3), which has 24 links. The traffic distribution for this network was taken from [6]. The results obtained by executing both our heuristic (called the Min-Hops heuristic) as well as the heuristic proposed in [4] are given in Table I.

As can be seen from the results, our heuristic not only performed as well as the heuristic in [4] in terms of minimizing NWR, but also did much better in minimizing the number of hops. This is due to the fact that our heuristic starts off by assigning each lightpath to the minimum hop path and then performs efficient rerouting such that the number of hops on any lightpath can increase by atmost 2 in a single iteration. As the same set of ligthpaths were setup by both heuristics, the total number of hops is an equivalent measure of the average number of hops, which is the average weighted hop count. Thus, the results substantiate our claim that our heuristic achieves the combined objective of minimizing network cost as well as maximizing resource utilization.

V. CONCLUSION

We considered the problem of Routing and Wavelength Assignment (RWA) in Virtual-Wavelength-Path (VWP) routed networks and took up the novel approach of not only minimizing the network cost, in terms of number of wavelengths and number of wavelength converters, but also maximizing the resource utilization, measured by the average weighted hop count. We proposed a heuristic algorithm for routing which not only tries to minimize the number of wavelengths required (NWR) but also minimizes the average number of hops taken up by a lightpath. We also presented a wavelength assignment procedure which minimizes the number of wavelength converters required. We compared our algorithm with one of the standard algorithms for this problem [4], and found the results to be highly encouraging.

In the future, we plan to tackle the same problem bringing the issue of survivability into consideration. Also, instead of using average weighted hop count as the measure for resource utilization, some other standard measures can be considered, which will put forward the need for different heuristics.

REFERENCES

[1] Rajiv Ramaswami and Kumar N. Sivarajan, "Routing and Wavelength Assignment in All-Optical Networks," *IEEE/ACM Transactions on Networking*, Vol. 3, No. 5, pp. 489–500, Oct. 1995.

[2] D. Banerjee and B. Mukherjee, "A Practical Approach for Routing and Wavelength Assignment in Large Wavelength-Routed Optical Networks," *IEEE Journal on Selected Areas in Communication*, Vol. 14, No. 5, pp. 903–908, Jun. 1996.

[3] Ken-ichi Sato, Satoru Okamoto and Hisaya Hadama, "Network Performance and Integrity Enhancement with Optical Path Layer Technologies," *IEEE Journal on Selected Areas in Communications*, Vol. 12, No. 1, pp. 159–170, Jan. 1994.

[4] N. Nagatsu, Y. Hamazumi and K. I. Sato, "Number of Wavelengths Required for Constructing Large-Scale Optical Path Networks," *Electronics and Communications in Japan, Part I - Communications*, Vol. 78, No. 9, 1995.

[5] Navarro Varela and Mark Sinclair, "Ant Colony Optimisation for Virtual-Wavelength-Path Routing and Wavelength Allocation," in *Proc. of the Congress on Evolutionary Computation (CEC'99)*, Washington DC, USA, pp. 1809-1816, July 1999.

[6] L. G. Tan and M. C. Sinclair, "Wavelength Assignment between the Central Nodes of the COST 239 European Optical Network," in *Proc. of 11th UK Performance Engineering Workshop*, Liverpool, pp. 235–247, Sep. 1995.

[7] Nico Wauters and Piet Demeester, "Design of the Optical Path Layer in Multiwavelength Cross-Connected Networks," *IEEE Journal on Selected Areas in Communication*, Vol. 14, No. 5, pp. 881–892, Jun. 1996.

[8] Massimo Tornatore, Guido Maier and Achille Pattavina, "WDM Network Optimization by ILP Based on Source Formulation," 2002.

[9] Andrea Fumagalli, Isabella Cerutti, Marco Tacca, Francesco Masetti, Rajesh Jagannathan and Sridhar Alagar, "Survivable networks based on optimal routing and WDM self-healing rings," in *Proc. IEEE INFOCOM 1999*, No. 1, pp. 726–733, Mar. 1999.

[10] Y. Miyao and H. Saito, "Optimal Design and Evaluation of Survivable WDM Transport Networks," *IEEE Journal on Selected Areas in Communications*, Vol. 16, No. 7, pp. 1190-1198, Sep. 1998.

[11] Rajiv Ramaswami and Kumar N. Sivarajan, "Optical Networks : A Practical Perspective," *Morgan Kaufmann Publishers*, 1998.