# A Wavelength Assignment Heuristic to Minimize SONET ADMs in WDM rings[*]

Xin Yuan  Amit Fulay
Department of Computer Science
Florida State University
Tallahassee, FL 32306
{xyuan, fulay}@cs.fsu.edu

## Abstract

*Optical Wavelength Division Multiplexing (WDM) rings are being deployed to support multiple SONET/SDH self-healing rings over a single physical optical ring. In such systems, the dominating cost is the SONET Add/Drop Multiplexers (ADMs). To minimize the system cost, algorithms must be developed to assign wavelengths to lightpaths in the system so that the number of ADMs required is minimized. However, the problem of optimal wavelength assignment to minimize SONET ADMs is NP–hard. Existing heuristic algorithms for this problem include the* assign first, *the* iterative matching *and the* iterative merging *heuristics. In this paper, we propose a new wavelength assignment heuristic to minimize SONET ADMs. Our heuristic is on average 3% to 5% more effective in finding the opportunities to share ADMs (and thus to reduce the total number of ADMs required) than the most effective existing heuristic, the iterative merging algorithm.*

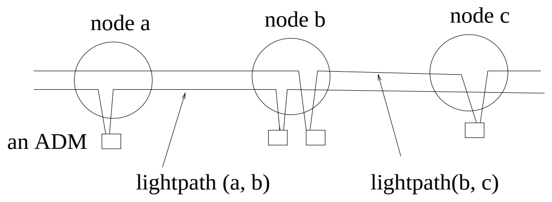
## 1 Introduction

Optical Wavelength Division Multiplexing (WDM) rings are being deployed to support multiple SONET/SDH self-healing rings over a single physical optical ring. One of the fundamental design problems for such networks is how to assign wavelengths to the lightpaths in the system so as to minimize the system cost. Since the system cost is dominated by the number of SONET Add/Drop Multiplexers (ADMs)[3, 4], we must develop effective wavelength assignment algorithms to minimize the number of SONET ADMs in the system.

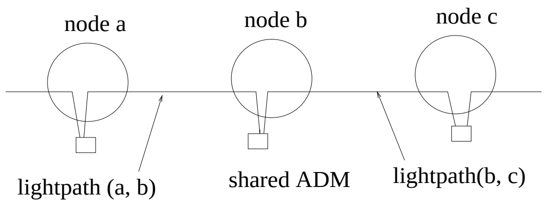
In a WDM ring supporting multiple SONET/SDH rings, the SONET ADMs are used to terminate lightpaths. Each lightpath uses two ADMs, one at each side of the lightpath. Although the origin node only needs the downstream ADM function and the termination node only needs the upstream ADM function, full ADMs are installed on both nodes to complete the protection path around the ring. Each wavelength around the ring provides the connectivity for a single SONET ring. Two adjacent lightpaths that are assigned the same wavelength can share an ADM at the common node. Figure 1 shows an example of ADM sharing. In the figure, we use the notion $(s, t)$ to represent a lightpath from node $s$ to node $t$. Figure 1 (a) depicts the case when lightpath $l_1 = (a, b)$ and lightpath $l_2 = (b, c)$ are assigned different wavelengths. In this case, 4 ADMs are needed to support the two lightpaths. Figure 1 (b) depicts the case when $l_1$ and $l_2$ are assigned the same wavelength, only 3 ADMs are needed. The ADM at node $b$ is shared by both lightpaths. Thus, as shown in the example, the wavelength assignment of the lightpaths directly affects the number of SONET ADMs needed in the system. Notice that the wavelength assignment problem has been extensively studied [1, 2, 5]. However, most of the existing wavelength assignment algorithms have a different optimization objective, that is, to minimize the total number of wavelengths required in the system. Thus, these algorithms cannot be directly applied to solve the problem of minimizing the number of SONET ADMs and new algorithms must be developed.

This paper studies effective wavelength assignment algorithms that minimize the number of SONET ADMs in the system. It has been shown in [6] that the optimal wavelength assignment for lightpaths to minimize SONET ADMs is NP-hard. Heuristic algorithms to deal with this problem, including *Cut–First*[3], *Assign–First*[3], *Iterative Merging*[6] and *Iterative Matching*[6], have been developed. All the existing algorithms use some kind of greedy heuristic to find the places where ADMs can be shared. Among all the existing heuristics, the iterative merging algorithm has been shown to be the most effective heuris-

[*]This project was supported in part by NSF grants: CCR-9904943 and CCR-0073482.



(a) lightpaths $(a, b)$ and $(b, c)$ are assigned different wavelengths, 4 ADMs are needed.



(b) lightpaths $(a, b)$ and $(b, c)$ are assigned the same wavelength, 1 ADM is shared.

**Figure 1. An example of sharing ADMs**

tic [6]. In this paper, we propose a wavelength assignment algorithm. Our algorithm is different from the existing heuristics in that (1) our algorithm explicitly attempts to find wavelength assignments so that lightpaths can form circles (as will be discussed later, forming lightpath circles is more effective in sharing ADMs than forming non–circles) and (2) our algorithm uses a heuristic called *least interference heuristic* to find more lightpaths that can share ADMs. By using these techniques, our algorithm is on average $3\%$ to $5\%$ more effective in reducing the number of ADMs required in the system compared to the existing most effective heuristic, the iterative merging algorithm.

The rest of the paper is structured as follows. Section 2 introduces the notations and the assumptions in this paper. Section 3 presents the existing heuristics. Section 4 describes our new algorithm. Section 5 reports the performance study. Section 6 concludes the paper.

## 2 Notations and assumptions

Given an N-node WDM ring network with the nodes labeled from 0 to $N-1$ and a set of full–duplex lightpaths, $S = \{(s_i, t_i)\}$, a wavelength assignment assigns a wavelength, $\lambda$, to each of the lightpaths in $S$. For a duplex lightpath $(s, t)$, we will call $s$ the origin node and $t$ the termination node. A wavelength assignment is *valid* if no two lightpaths that share a common link are assigned the same wavelength.

When two adjacent lightpaths $l_1 = (a, b)$ and $l_2 = (b, c)$ are assigned the same wavelength, an ADM can be shared in node $b$. The process of finding two lightpaths sharing an ADM is called *merging* the two lightpaths since once the two lightpaths, $l_1 = (a, b)$ and $l_2 = (b, c)$, share an ADM, the two lightpaths can be treated as one lightpath $l_{12} = (a, c)$. A *segment* contains one or more merged lightpaths such that the termination of a lightpath (except the last one) is the origin of the subsequent lightpaths and no two lightpaths share a common link. A segment is said to be a *circle* if the segment occupies the whole ring.

In this paper, we will focus on the *maximum ADMs sharing problem*, that is, finding a valid wavelength assignment scheme such that the number of shared ADMs is maximum. This process is done by merging lightpaths into segments. We will approach this problem with the following assumptions:

- We consider static wavelength assignment. The set of lightpaths to assign wavelengths is known a prior.
- We do not consider the routing issue in this paper. We will assume that a lightpath is routed clockwise on the ring. The previous work in this problem [3, 6] made the same assumption.
- We focus on minimizing the number of ADMs and assume that the number of wavelength is infinite. As pointed out in [3, 6], minimizing the number of ADMs and minimizing the number of wavelengths in the system can sometimes be contradictory.
- We assume that a lightpath cannot be split. Thus, the algorithm can only assign wavelengths to the lightpaths, but cannot change the lightpaths.

## 3 Existing heuristics

A number of wavelength assignment heuristics to minimize the number of SONET ADMs have been proposed. Some of them, such as the cut–first heuristic [3], assume that a lightpath can be split. In this paper, however, we will only consider the heuristics that work when the lightpaths cannot be split. The existing heuristics include the assign first, the iterative matching and the iterative merging heuristics. Next, we will describe these three heuristics.

### 3.1 Assign First

The assign first heuristic [3] takes advantage of the fact that there exists an efficient optimal algorithm for wavelength assignment to minimize the number of ADMs for a linear array topology. Since in a linear array, lightpaths do not wrap around and all adjacent lightpaths can be merged, using a greedy method to merge all possible adjacent lightpaths will result in a minimum number of ADMs required in the system.

Given this simple algorithm for linear arrays, the assign first heuristic tries to reduce the wavelength assignment problem for rings to the wavelength assignment problem on linear arrays by doing the following. First, the assign first algorithm carefully selects a link such that the number of lightpaths that pass through the link is minimum. Then, it assigns all the lightpaths that pass through the selected link with different wavelengths. After that, none of the remaining lightpaths can pass through the link and the greedy algorithm for linear arrays is used to assign wavelengths to the remaining lightpaths.

## 3.2 Iterative Matching

The iterative matching algorithm [6] works as follows. Initially, we have $|R|$ segments, with each segment consisting of one lightpath. At each step, a bipartite graph $G_i = (U_i, V_i, E_i)$ is constructed for each node $n_i$, where

- $U_i$ is the set of segments ending at node $n_i$.
- $V_i$ is the set of segments starting from node $n_i$.
- For any $u \in U_i$ and $v \in V_i$, $(u, v) \in E_i$ if and only if $u$ and $v$ do not overlap with each other, that is, segment $u$ can be merged with segment $v$.

The maximum matching of $G_i$ is then found. The node that results in the largest maximum matching is then merged. After two segments are merged, the combined segment will be treated as one segment and the same procedure will apply to the new segments until no more potential merges can be found. Since the maximum matching algorithm [7] for a bipartite graph runs in polynomial time, this algorithm is a polynomial time algorithm.

## 3.3 Iterative Merging

The iterative merging algorithm [6] works as follows. Initially, there are $|R|$ segments, each segment consisting of one lightpath. At each step, one of the following three possible operations is performed in decreasing order. This process continues until no more operations can be performed.

- *Operation 1.* Merge two noncircle segments into a circle segment.
- *Operation 2.* Split a noncircle segment into two noncircle segments and then merge one of them with another noncircle segment into a circle segment.
- *Operation 3.* Merge two noncircle segments into a larger noncircle segment.

Each of the three operations increases the number of shared ADMs. Operation 1 increases the number by two, while Operations 2 and 3 increase the number by one. For $|R|$ lightpaths, there can at most have $|R|$ shared ADMs. Thus, the algorithm terminates at most after $|R|$ steps.

## 3.4 Discussion

All the heuristics are polynomial time algorithms. Since a typical SONET ring contains 16 nodes with a few hundred lightpaths, almost all polynomial time algorithms can be considered as efficient algorithms since these algorithms will be executed only once. Thus, we will ignore the time complexity issue and focus on the algorithms' ability to find shared ADMs. The assign first heuristic optimally assigns wavelengths to the set of lightpaths that do not pass through the selected link. However, for a ring of 16 nodes, many lightpaths will pass any given link. The assign first heuristic does not have the ability to find any ADM sharing opportunities for these lightpaths. The iterative matching algorithm finds the best wavelength assignment for a selected node. It suffers from two drawbacks. First, merging lightpaths in one node may affect the merging of lightpaths in other nodes. Merging all possible lightpaths for a given node may not be the best heuristic. Second, the iterative matching algorithm does not consider merging lightpaths into circles. As will be discussed in the next section, merging lightpaths into circles is an effective way of sharing ADMs. The main advantage of the iterative merging is that it tries to merge lightpaths into circles. As shown in [6], the iterative merging algorithm is on average about 40% more effective than the assign first heuristic and about 10% more effective than the iterative matching heuristic. Thus, in this paper, we will focus on improving the best heuristic, the iterative merging algorithm.

## 4 A new wavelength assignment heuristic

A study of the wavelength assignment results produced by the three heuristics reveals that the major advantage of the iterative merging algorithm lies in that it forms more circle segments than the assign first and iterative matching heuristics. Both the assign first and the iterative matching heuristics do not take this important factor into consideration. For a circle that contains $k$ lightpaths, $k$ ADMs are needed and $k$ ADMs are shared. When merging $k$ lightpaths into a non-circle segment, $k - 1$ ADMs are shared. Table 1 shows the differences in terms of sharing ADMs when merging lightpaths into circle and non–circle segments. As can be seen in the table, forming a circle segment is a very effective way to share ADMs, especially when the circle contains a small number of lightpaths. For example, when

| No. of lightpaths | No. of shared ADMs | | difference |
|---|---|---|---|
| | non-circle | circle | |
| 2 | 1 | 2 | 100% |
| 3 | 2 | 3 | 50% |
| 4 | 3 | 4 | 33% |
| 5 | 4 | 5 | 20% |
| 6 | 5 | 6 | 16% |

**Table 1. Differences of merging lightpaths into a non-circle segment and into a circle segment**

merging 2 lightpaths, forming a circle is 100% more effective than forming a noncircle. For 3 lightpaths, forming a circle is 50% more effective. Thus, for a wavelength assignment algorithm to be effective in finding the opportunities for sharing ADMs, the algorithm must be able to find circles, especially the ones with a small number of lightpaths.

The iterative merging algorithm tries to merge segments into circles. It goes as far as to break some non-circle segments in order to form circles in Operation 2. However, when a circle contains more than 2 lightpaths, the iterative merging algorithm does not guarantee to find the circle. Operation 1 in the iterative merging algorithm only guarantees to find all circles that contain two lightpaths. In our algorithm, we propose to use a greedy breadth first search algorithm to find as many circles as possible before any merging of lightpaths takes place. Although finding the maximum number of circles can be difficult, the breadth first search algorithm can guarantee find a circle of any length in $O(|R|^2)$ time if such a circle exists. Here $|R|$ is the number of lightpaths. Since circles with a smaller number of lightpaths share ADMs more effectively as shown in Table 1. The breadth first search algorithm will first be used to find circles with 2 lightpaths, and then circles with 3 lightpaths, and so on until no more circles can be found.

In addition to finding as many circles as possible, our algorithm also uses the *least interference heuristic* to find more lightpaths that can share ADMs. Notice that in the iterative merging algorithm, when no circles can be formed, the algorithm will perform Operation 3, which merges segments the first time it finds such opportunity. In other words, the iterative merging algorithm does not use any heuristic to improve the merging chances. The least interference heuristic evaluates each merging opportunity and carefully chooses the order to merge segments in order to find more ADM sharing opportunities. The least interference heuristic works as follows. Given a set of segments, the heuristic finds all pairs of segments that can be merged. Each of such pairs can lead to a merging of segments (1 shared ADM). The heuristic will then compute a weight for each of the

```
Find_A_Circle(lightpath: startpath)
(1)Create a segment, $S$, containing the lightpath $startpath$.
(2)Insert $S$ into the $queue$
(3)**while** ($queue$ is not empty) **do**
(4)     $Seg = dequeue()$
(5)     Let $Seg.start$ be the starting node of Seg.
        Let $Seg.end$ be the ending node of $Seg$
(6)     **for** each lightpath $p$ that starts from $Seg.end$ **do**
(7)         **if** ($p$ and $Seg$ form a circle) **then**
(8)             return the circle
(9)         **else if** ($p$ can be merged with $Seg$) **then**
(10)            **if** ($p.end$ is not marked) **then**
(11)                insert $p + Seg$ into the queue
(12)            **end if**
(13)        **end if**
(14)    **end for**
(15)    Mark $Seg.end$
(16)**end while**
(17)return no more circles
```

**Figure 2. The breadth first algorithm to find a circle**

pairs. The weight of a pair $p$ is equal to the number of pairs that can be merged assuming that the $p$ has been merged. Hence, the weight of a pair $p$ is the number of potential merging opportunities after $p$ is merged. The heuristic will then merge the pair with the maximum weight. Thus, the heuristic always selects to merge the pair that will have the least interference with the rest of the merging opportunities. This is why the heuristic is called the least interference heuristic. By merging the least interference pairs first, it is likely that the heuristic will find more lightpaths that can share ADMs.

Figure 2 shows the breadth first search algorithm to find a circle that starts from a given lightpath. The algorithm takes the lightpath as a parameter and determines if there is a circle that can be formed starting from the lightpath. This algorithm can easily be modified to find circles that contain a certain number of lightpaths. The worst case time complexity of the algorithm is $O(|R|)$. To determine whether there is a circle starting from any lightpath, $O(|R|^2)$ time is needed. Figure 3 shows the new heuristic. The first 5 lines use the greedy algorithm to find circles. Lines (6) to (10) is the least interference heuristic. Since a circle can be found in $O(|R|^2)$ time, the time complexity for lines (1) to (5) is $O(|R|^3)$. The while loop in line (6) executes at most $|R|$ times since in each iteration, at least one shared ADM is found. Lines (7) and (8) have the worst case time complexity of $O(|R|^3)$. Thus, the time complexity of the whole algorithm is $O(|R|^4)$.

```
New_Wavelength_Assignment_Heuristic
(1)   For $i = 2$ to ring_size do
(2)       While (there exists a circle of $i$ lightpaths) do
(3)           Merge the $i$ lightpaths into a circle.
(4)       End While
(5)   End For
(6)   While (there exist more merging opportunities) do
(7)       Find all potential merging pairs of segments
(8)       Compute the weight for each pair
(9)       Merge the pair with the largest weight
(10)  End While
```

**Figure 3. The new wavelength assignment algorithm**

## An example

Let us use an example to show how the proposed algorithm works. Consider wavelength assignment for the set of lightpaths: T={(0, 1), (1, 2), (0, 2), (2, 4), (1, 3), (3, 4), (4, 5), (5, 6), (5, 6), (6, 4), (6, 5)}
on an 8-node ring. The heuristic will first find the a circle with 2 lightpaths, $\{(5,6),(6,5)\}$ and a circle with 3 lightpaths $\{(4,5),(5,6),(6,4)\}$. Since there are no more circles can be formed. The algorithm will use the least interference heuristic to merge the rest of the lightpaths:

$$\{(0,1),(1,2),(0,2),(2,4),(1,3),(3,4)\}$$

For this set of segments, there are five potential pairs of segments that can be merged. The potential pairs and their weights are as follows:

$$weight((0,1),(1,2)) = 3$$
$$weight((0,1),(1,3)) = 3$$
$$weight((0,2),(2,4)) = 3$$
$$weight((1,2),(2,4)) = 3$$
$$weight((1,3),(3,4)) = 4$$

Here, the weight of pair $((1,3),(3,4))$ is computed by calculating the number of potential merges assuming that this pair is merged. In this case, we still have four potential merges $(0,1)$ with $(1,2)$, $(0,1)$ with $(1,3)$, $(0,2)$ with $(2,4)$ and $(1,2)$ with $(2,4)$. Thus, $weight((1,3),(3,4)) = 4$. Since $weight((1,3),(3,4))$ is the largest, the heuristic will select to merge $(1,3)$ and $(3,4)$. After this merging the same process will be applied to the set of segments:

$$\{(0,1),(1,2),(0,2),(2,4),(1,4)\}$$

Notice that the segments $(1,3)$ and $(3,4)$ is replaced by their combined segment $(1,4)$. The pair $((0,1),(1,2))$ will be merged next, resulting the remaining set of $\{(0,2),(0,2),(2,4),(1,4)\}$. At last, the pair $((0,2),(2,4))$ will be merged and the algorithm terminates. Thus, for this example, the total number of shared ADMs is 8 and the final wavelength assignment is shown in Figure 4.

| lightpaths | IMerge | our algo. | improvement |
|---|---|---|---|
| 50 | 22.5 | 23.5 | 4.4 |
| 75 | 39.1 | 40.8 | 4.3 |
| 100 | 56.4 | 58.7 | 4.1 |
| 125 | 75.5 | 78.3 | 3.7 |
| 150 | 95.0 | 98.1 | 3.3 |

**Table 2. Improvement of our new iterative merging algorithm**

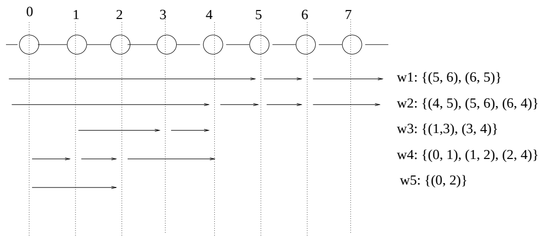


**Figure 4. Final wavelength assignment for the example**

## 5 Performance study

In this section, we compare the performance of the Iterative Merging algorithm (IMerge) with our proposed heuristic. We do not compare our heuristic with other existing heuristic since in [6], it has been shown that the Iterative merging algorithm performs significantly better than other heuristics. We study the performance through simulation. The underlying ring network consists of 16 nodes (16 is recommended to be the maximal number of nodes for SONET rings).

Table 2 shows the improvement resulting from our heuristic in comparison of the iterative merging algorithm. The number of ADMs shared shown in the figure are the average of 1000 experiments. The second column shows the number of shared ADMs found using the iterative merging algorithm. The third column shows that number of shared ADMs found using our new algorithm. The last column shows the improvement percentage. As can be seen in the table, our algorithm is on average 3% to 5% more effective than the iterative merging algorithm in finding the shared ADMs. This improvement is consistent regardless of the number of lightpaths in the system. Notice that the complexity of our heuristic is not significantly larger than that of the iterative merging algorithm and thus, the improvement is obtained without much additional computational costs.

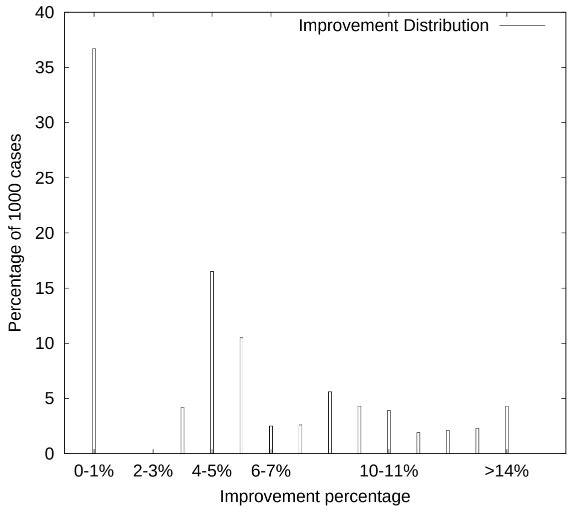


**Figure 5. The improvement distribution**

Figure 5 is the histogram about the improvement for 1000 experiments with 50 lightpaths in a 16-node ring. As can be seen in the figure, for many cases (about $36\%$ of all cases) there is no improvement using the new heuristic. This is because the number of lightpaths is small and both heuristics result in the same wavelength assignment. However, for other cases, the improvement using the new heuristic is fairly large. Although the average improvement is $4.7\%$, for about $14\%$ of all cases, our new algorithm results in more than $10\%$ improvement. It should be noted that among all the 1000 experiments, there is not a single case that the new heuristic performs worse than the iterative merging algorithm. Figure 6 shows that histogram for the experiments with 100 lightpaths. In this case, the improvement is more evenly distributed from $0\%$ to $10\%$. We observe that for a small fraction of cases, (about $2\%$ of all cases) our algorithm perform worse than the iterative merging algorithm. However, when our algorithm performs worse than the iterative merging algorithm, it is only worse by a small percentage (less than $2\%$). Experiments with other numbers of lightpaths yield similar results. All these indicate that our heuristic is indeed more effective than the iterative merging algorithm.

## 6 Conclusion

When WDM rings are used to support multiple SONET/SDH self-healing rings over a single physical optical ring, effective wavelength assignment algorithms must be developed to minimize the number of SONET ADMs in order to minimize the system cost. In this paper, we propose a new wavelength assignment heuristic that minimizes the number of SONET ADMs. Our algorithm is on average $3\%$ to 5% more effective than the existing, most effective wavelength assignment heuristic, the iterative merging heuristic. In addition, our algorithm does not introduce much extra computational overheads. We are currently investigating applying the integer programming and simulation annealing techniques to find better solutions for this problem.

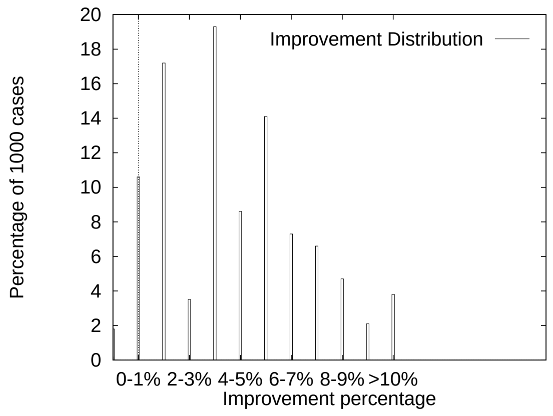


**Figure 6. The improvement distribution**

## References

[1] R. Barry and P. Humblet, “Models of Blocking Probability in All-Optical Networks with and without Wavelength Changers”, *IEEE JSAC: Special Issue on Optical Networks*, vol. 15, no. 5, pages 858-867, 1996.

[2] I. Chlamtac, A. Ganz and G. Karmi, “Lightnets: Topologies for High–Speed Optical Networks”, *IEEE/OSA Journal on Lightwave Technology*, vol. 11, pages 951–961, 1993.

[3] O. Gerstel, P. Lin and Sasaki “Wavelength Assignment in a WDM Ring to Minimize Cost of Embedded SONET Rings”, *IEEE INFOCOM*, pages 94-101, 1998.

[4] O. Gerstel, G. Sasaki and R. Ramaswami, “Cost Effective Traffic Grooming in WDM Rings”, *IEEE INFOCOM*, pages 69-77, 1998.

[5] M. Kovacevic and A. Acampory, “Benefits of Wavelength Translation in Alll Optical Clear-Channel Networks”, *IEEE JSAC: Special Issues on Optical Networks*, vol 14, no. 5, pages 868-880, 1996.

[6] L. Liu, X. Li, P. Wan and O. Frieder, “Wavelength Assignment in WDM Rings to Minimize SONET ADMs”, *IEEE INFOCOM*, 2000.

[7] Udi Manber, “Introduction to Algorithm: A Creative Approach”, Addison-Wesley, 1989.