Wavelength Assignment to Minimize the Number of 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 SONET/SDH self-healing rings. In such systems, multiple SONET/SDH self-healing rings are realized over a single physical optical ring through wavelength division multiplexing. The cost of such a system is dominated by 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. This problem of optimal wavelength assignment to minimize the number of SONET ADMs is known to be NP–hard. Existing heuristic algorithms for this problem include the *assign first* heuristic, the *iterative matching* heuristic and the *iterative merging* heuristic. In this paper, we develop an integer linear programming (ILP) formation for this problem, propose a new wavelength assignment heuristic, and evaluate the existing and the newly proposed heuristic using the ILP formation. We conclude that the performance of the newly proposed heuristic is very close to optimal.

Keywords: Wavelength Assignment, WDM, SONET.

*A preliminary version of this paper is published in the IEEE International Conference on Communications 2002.

†Preferred Address: Xin Yuan, 203 Love Building, Department of Computer Science, Florida State University, Tallahassee, FL 32306, USA. Phone: (850)644-9133. Fax: (850)644-0058. Email: xyuan@cs.fsu.edu.

List of Figures

List of Tables

1 Introduction

Optical Wavelength Division Multiplexing (WDM) rings are being deployed to support SONET/SDH self-healing rings. In such systems, multiple SONET/SDH self-healing rings are realized over a single physical optical ring through wavelength division multiplexing. 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 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 end 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. Fig. 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. Fig. 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. Fig. 1 (b) depicts the case when l_1 and l_2 are assigned the same wavelength. In this case, the ADM at node b is shared by both lightpaths and only 3 ADMs are needed. This example shows that wavelength assignment 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. These algorithms cannot be directly applied to solve the problem of minimizing the number of SONET ADMs and new algorithms must be developed.

It has been shown in [6] that the optimal wavelength assignment problem to minimize the number of SONET ADMs is NP-hard. Heuristic algorithms to solve this problem include *Cut–First*[3], *Assign–First*[3], *Iterative Merging*[6] and *Iterative Matching*[6]. While the relative performance of these heuristics has been studied in [3, 6], it is unclear how these heuristics perform with respect to the optimal solutions. In this paper, we develop an integer linear programming (ILP) formation for this problem, propose a new wavelength assignment heuristic that improves over the existing most effective heuristic, and evaluate the existing and the newly proposed wavelength assignment heuristics using the ILP formation. We conclude that the performance of the newly proposed wavelength assignment heuristic is close to that of the optimal algorithm.

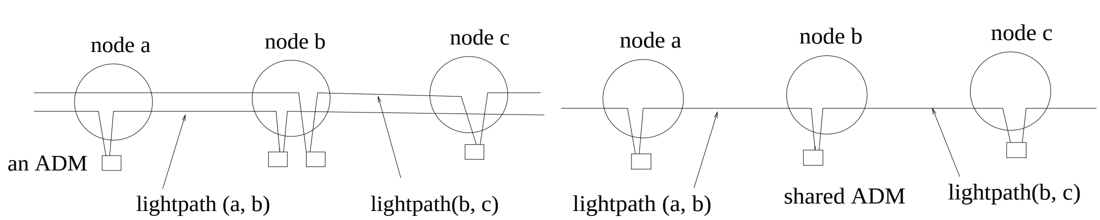
The rest of the paper is structured as follows. Section 2 introduces the notations and the assumptions and briefly describes the existing wavelength assignment heuristics. Section 3 presents the ILP formation. Section 4 describes our heuristic. Section 5 gives an example. Section 6 reports the results of the performance study. Section 7 concludes the paper.

2 Background

2.1 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, $R = \{(s_i, t_i)\}$, a wavelength assignment assigns a wavelength, λ, to each of the lightpaths in R. 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 merged 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



(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

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 ADM sharing problem*, that is, finding a valid wavelength assignment scheme such that the number of shared ADMs is maximum. The wavelength assignment to maximize ADM sharing can be solved in two phases. In the first phase, individual lightpaths are merged into segments such that the number of shared ADMs is maximum. In the second phase, wavelengths are assigned to the segments. Since the second phase only affects the number of wavelengths used, but not the number of shared ADMs, this paper will focus on the first phase. 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.

2.2 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. Next, we will briefly describe the existing heuristics, including the assign first heuristic [3], the iterative matching heuristic [6] and the iterative merging heuristic [6].

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 maximum number of shared ADMs.

Given this simple algorithm for linear arrays, the assign first heuristic reduces the wavelength assignment problem for rings to the wavelength assignment problem for 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 applied to assign wavelengths to the remaining lightpaths.

Iterative Matching

The iterative matching algorithm [6] works as follows. Let R be the set of lightpaths to be realized. 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 [8] for a bipartite graph runs in polynomial time, this algorithm is a polynomial time algorithm.

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.

The performance of these heuristics was compared in [6]. It was shown that 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 [6]. However, it is not clear how the heuristics perform with respect to optimal solutions. In this paper, we develop an integer linear programming (ILP) formation for the wavelength assignment optimization problem and evaluate the performance of the heuristics. We further develop a wavelength assignment algorithm that improves over the best existing heuristic, the iterative merging algorithm. Our results show that the performance of the newly developed heuristic is very close to that of the optimal algorithm. Next, we will first present the integer linear programming formation for the wavelength assignment optimization problem, and then describe our new wavelength assignment heuristic.

3 The ILP formation

An integer linear programming (ILP) formation that is based on the integral multi-commodity model was presented in [6]. However, this ILP formation introduces too many variables and constraints even for a small sized problem. In this section, we develop an ILP formation that allows us to obtain optimal solutions for reasonable sized problems.

Let the nodes in an N-node WDM ring network be labeled from 0 to $N-1$. Let the set of lightpaths be $R = \{l_1, l_2, ..., l_{|R|}\}$. A lightpath $l_i = (s_i, t_i)$ can be merged with another lightpath $l_j = (s_j, t_j)$ if and only if the following two conditions are satisfied, (1) $t_i = s_j$, and (2) l_i and l_j do not share any link.

A variable $v(l_i, l_j)$ is created for each pair of lightpaths (l_i, l_j) that can be merged. The solution for $v(l_i, l_j)$ determines whether l_i should be merged with l_j. $v(l_i, l_j) = 1$ indicates that l_i is merged with l_j, and $v(l_i, l_j) = 0$ indicates that l_i and l_j are not merged. Notice that if two lightpaths l_i and l_j can form a circle, two variables $v(l_i, l_j)$ and $v(l_j, l_i)$ are created. Since each merge results in one shared ADM, the sum of the variables $v(l_i, l_j)$ is the total number of ADMs shared. Thus, the objective function of the ILP is to

$$maximize \sum_{for \;\; all \;\; v(l_i, l_j)} \{v(l_i, l_j)\}.$$

The objective function is optimized under the following constraints.

First, the values for all $v(l_i, l_j)$'s must be either 0 or 1.

$$0 \le v(l_i, l_j) \le 1 \;\; and \;\; v(l_i, l_j) \;\; is \;\; an \;\; integer$$

Second, an endpoint of a lightpath can be merged with another lightpath at most once.

$$\sum_{l_i \in R} v(l_i, l_j) \le 1$$

$$\sum_{l_j \in R} v(l_i, l_j) \le 1$$

Third, a segment should not contain lightpaths that overlap. To ensure this, we consider all potential segments. Let a potential non-circle segment, NC, contain n lightpaths $l_1 = (s_1, t_1)$, $l_2 = (t_1, t_2)$, ..., and $l_n = (t_{n-1}, t_n)$. Let $o_1 = (t_n, to_1)$, $o_2 = (t_n, to_2)$, ..., and $o_k = (t_n, to_k)$, be *all* the k lightpaths that can be merged with l_n, but overlap with other lightpaths in NC. For each potential non-circle segment, NC, the following constraint must be satisfied.

$$\sum_{i=1}^{n-1} v(l_i, l_{i+1}) + \sum_{j=1}^{k} v(l_n, o_j) \le n - 1$$

This constraint ensures that when lightpaths l_1, l_2, ..., and l_n, are merged into a segment ($v(l_i, l_{i+1}) = 1$ for all $i = 1..n-1$ and $\sum_{i=1}^{n-1} v(l_i, l_{i+1}) = n - 1$), l_n will not merge with any of the lightpaths o_1, o_2, ..., and o_k ($v(l_n, o_j) = 0$ for all $j = 1..k$). Notice that the number of this type of constraints can be exponential with respect to the number of lightpaths.

Fourth, for an n-lightpath circle segment that contains lightpaths $l_1 = (s_1, t_1)$, $l_2 = (t_1, t_2)$, ..., and $l_n = (t_{n-1}, s_1)$, the merges of $n - 1$ merging points in the circle implies that all the n lightpaths are in the same segment and that the nth merging point is also merged. To ensure this in the wavelength assignment solution, for each potential n-lightpath circle segment that contains lightpaths $l_1 = (s_1, t_1)$, $l_2 = (t_1, t_2)$, ..., and $l_n = (t_{n-1}, s_1)$, the following n constraints must be satisfied.

$$-v(l_1, l_2) + v(l_2, l_3) + ... + v(l_{n-1}, l_n) + v(l_n, l_1) \leq n - 2$$

$$v(l_1, l_2) - v(l_2, l_3) + ... + v(l_{n-1}, l_n) + v(l_n, l_1) \leq n - 2$$

$$...$$

$$v(l_1, l_2) + v(l_2, l_3) + ... - v(l_{n-1}, l_n) + v(l_n, l_1) \leq n - 2$$

$$v(l_1, l_2) + v(l_2, l_3) + ... + v(l_{n-1}, l_n) - v(l_n, l_1) \leq n - 2$$

For example, consider three lightpaths $l_1 = (1, 3)$, $l_2 = (3, 6)$, and $l_3 = (6, 1)$. These three lightpaths can potentially form a circle. We will generate three constraints for the potential circle.

$$-v(l_1, l_2) + v(l_2, l_3) + v(l_3, l_1) \leq 1$$

$$v(l_1, l_2) - v(l_2, l_3) + v(l_3, l_1) \leq 1$$

$$v(l_1, l_2) + v(l_2, l_3) - v(l_3, l_1) \leq 1$$

Basically, any two of the three values $v(l_1, l_2)$, $v(l_2, l_3)$, and $v(l_3, l_1)$ equal to 1 implies that the three lightpaths are in the same segment (circle) and the third value should also be 1.

A solution to this ILP formation can determine the segments and the maximum number of shared ADMs. Following is a lemma that can be applied to reduce size of the wavelength assignment problem.

Lemma 1: Given an N-node WDM ring network with the nodes labeled from 0 to $N - 1$. Let the set of lightpaths, R, contain two lightpaths $l_1 = (s, t)$ and $l_2 = (t, s)$. There exists an optimal wavelength assignment such that l_1 and l_2 are assigned to the same wavelength.

Proof: When l_1 and l_2 are assigned the same wavelength, they form a 2-lightpath circle. Assuming that in an optimal wavelength assignment scheme, l_1 and l_2 are assigned different wavelengths. Let segment S_1 contain l_1 and segment S_2 contain l_2. As shown in Fig. 2, S_1 can be one of the four possible forms (1), (2), (3) and (4), and S_2 can be one of the four possible forms (5), (6), (7) and (8). In the figure, a, b, c and d may contain one or multiple merged lightpaths.

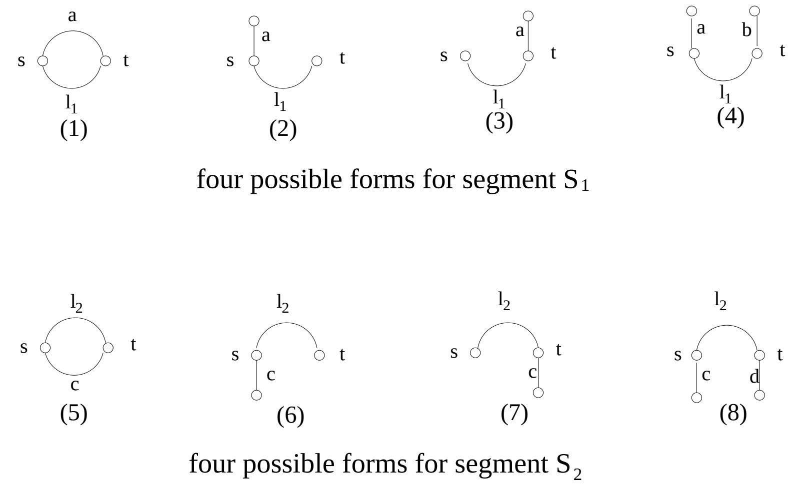


Figure 2: Possible forms for segments S_1 and S_2

We can obtain a new wavelength assignment scheme by reassigning wavelengths for the lightpaths in S_1 and S_2. Specifically, we will assign the same wavelength to l_1 and l_2 and try to merge the rest of the lightpaths in S_1 and S_2. It can be shown that for all 16 possible combinations of the forms for S_1 and S_2, the new wavelength assignment scheme is also optimal. Here we will show one case: S_1 is of the form (1) and S_2 is of the form (5). In this case, taking l_1 out of S_1 and l_2 out of S_2 results in a reduction of 4 shared ADMs. However, the merging of l_1 and l_2 results in 2 shared ADMs and the merging of a and c results in another 2 shared ADMs. Thus, the new wavelength assignment scheme is also optimal. Consider all the 16 cases, it can be proven that the new wavelength assignment scheme shares as many ADMs as the optimal wavelength assignment scheme. Thus, there exists an optimal wavelength assignment scheme where l_1 and l_2 are assigned the same wavelength. $\square$

This lemma indicates that to find optimal wavelength assignment for a set of lightpaths, we can first find all 2-lightpath circles in the set of lightpaths and assign different wavelengths to all 2-lightpath circles, and then find the optimal wavelength assignment for the rest of the lightpaths. The optimality does not hold for 3-lightpath circles. Consider a counter example for an 8 nodes ring with $R = \{(0,3),(3,5),(5,0),(0,1),(1,5),(5,6),(6,3)\}$. Lightpaths $(0,3)$, $(3,5)$ and $(5,0)$ form a 3-lightpath circle. A wavelength assignment with this circle can at most share 5 ADMs since merging lightpaths $(0,1)$, $(1,5)$, $(5,6)$, $(6,3)$ can share at most 2 ADMs (to share 3 ADMs would require all four lightpaths to be merged into one segment, which is impossible). Thus, merging lightpaths $(0,3)$, $(3,5)$ and $(5,0)$ into a circle yields a sub-optimal solution since there exists another wavelength

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	25%
6	5	6	20%

Table 1: Differences in merging lightpaths into a non-circle segment and into a circle segment

assignment scheme that shares 6 ADMs by merging lightpaths $(5,0)$, $(0,1)$ and $(1,5)$ into a circle and lightpaths $(3,5)$, $(5,6)$ and $(6,3)$ into another circle.

4 A new wavelength assignment heuristic

In this section, we propose a new wavelength assignment heuristic to minimize SONET ADMs. Our heuristic is different from the existing heuristics in that (1) our algorithm explicitly attempts to find as many circle segments as possible, and (2) our algorithm uses a heuristic called the *least interference heuristic* to find more lightpaths that can share ADMs.

The idea behind finding as many circle segments as possible is that forming circle segments is more effective in sharing ADMs than forming non-circle segments. Forming a k-lightpath circle shares k ADMs while forming a k-lightpath non-circle segment only shares $k-1$ ADMs. 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 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. Notice that the iterative merging heuristic also tries to merge segments into circles. However, when a circle contains more than 2 lightpaths, the iterative merging algorithm does not guarantee to find that circle.

We propose to use a greedy breadth first search algorithm to find as many circles as possible before any other merging of lightpaths takes place. Although finding the maximum number of circles can be difficult, the breadth

Find_A_Circle(lightpath: startpath)

(1) Create a segment, S, containing $startpath$.

(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 3: The breadth first algorithm to find a circle

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 4: The new wavelength assignment algorithm

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. Fig. 3 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 that lightpath. This algorithm can easily be modified to find a circle that contains a given 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.

After circles are found by the breadth first search algorithm, our algorithm also uses the *least interference heuristic* to determine the order of further lightpath merging. The least interference heuristic evaluates each merging opportunity and carefully chooses the order to merge segments so as 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 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 pair first, it is likely that the heuristic will find more lightpaths that can share ADMs.

Fig. 4 shows the new heuristic. The first 5 lines use the greedy breath first search algorithm to find circles. Since circles with a smaller number of lightpaths share ADMs more effectively, the algorithm tries to find circles with fewer number of lightpaths first, that is, it first finds circles with 2 lightpaths, and then circles with 3 lightpaths, and so on until no more circles can be found. Lines (6) to (10) realize 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 $O(|R|^4)$.

5 An example

Consider wavelength assignment for the set of lightpaths in an 8-node ring: $R = \{l_1 = (0, 2), l_2 = (2, 4), l_3 = (1, 3), l_4 = (3, 4), l_5 = (4, 5), l_6 = (5, 6), l_7 = (6, 4), l_8 = (6, 5)\}$.

The ILP formation

In the ILP formation, the following 9 variables will be created: $v(l_1, l_2)$, $v(l_2, l_5)$, $v(l_3, l_4)$, $v(l_4, l_5)$, $v(l_5, l_6)$, $v(l_6, l_7)$, $v(l_6, l_8)$, $v(l_7, l_5)$, $v(l_8, l_6)$.

The ILP formation is to maximum the sum of these variables with the following constraints.

(1) constraints to ensure the value of each variable to be either 0 or 1.

$$0 \leq v(l_1, l_2),\ v(l_2, l_5),\ v(l_3, l_4),\ v(l_4, l_5),\ v(l_5, l_6) \leq 1$$

$$0 \leq v(l_6, l_7),\ v(l_6, l_8),\ v(l_7, l_5),\ v(l_8, l_6) \leq 1$$

(2) constraints to ensure an endpoint of a lightpath to be merged with another lightpath at most once.

$$v(l_6, l_7) + v(l_6, l_8) \leq 1$$

$$v(l_2, l_5) + v(l_4, l_5) + v(l_7, l_5) \leq 1$$

$$v(l_5, l_6) + v(l_8, l_6) \leq 1$$

(3) constraints to ensure that the lightpaths in a segment do not overlap.

$$v(l_1, l_2) + v(l_2, l_5) + v(l_5, l_6) + v(l_6, l_7) + v(l_6, l_8) \leq 3$$

$$v(l_2, l_5) + v(l_5, l_6) + v(l_6, l_7) + v(l_6, l_8) \leq 2$$

$$v(l_3, l_4) + v(l_4, l_5) + v(l_5, l_6) + v(l_6, l_7) + v(l_6, l_8) \leq 3$$

$$v(l_4, l_5) + v(l_5, l_6) + v(l_6, l_7) + v(l_6, l_8) \leq 2$$

$$v(l_5, l_6) + v(l_6, l_8) \leq 1$$

(4) constraints for circle segments two potential circles formed by l_5, l_6 and l_7, and by l_6 and l_8.

$$-v(l_5, l_6) + v(l_6, l_7) + v(l_7, l_5) \leq 1$$

$$v(l_5, l_6) - v(l_6, l_7) + v(l_7, l_5) \leq 1$$

$$v(l_5, l_6) + v(l_6, l_7) - v(l_7, l_5) \leq 1$$

$$v(l_8, l_6) - v(l_6, l_8) \leq 0$$

$$-v(l_8, l_6) + v(l_6, l_8) \leq 0$$

Solving this ILP problem with lp_solve [7], we obtain the values for $v(l_1, l_2)$, $v(l_3, l_4)$, $v(l_4, l_5)$, $v(l_6, l_8)$, and $v(l_8, l_6)$, are 1's. Thus, the optimal solution for this problem is to have 5 shared ADMs and there are four segments, $\{l_1, l_2\}$, $\{l_3, l_4, l_5\}$, $\{l_6, l_8\}$, and $\{l_7\}$.

The proposed heuristic

The proposed heuristic will first find circles. In this example, it will find a 2-lightpath circle consisting of l_6 and l_8. After that, the algorithm will use the least interference heuristic to merge the rest of the lightpaths, l_1, l_2, l_3, l_4, l_5, and l_7.

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: $weight(l_1, l_2) = 4$, $weight(l_2, l_5) = 2$, $weight(l_3, l_4) = 4$, $weight(l_4, l_5) = 2$, and $weight(l_7, l_5) = 2$. Here, the $weight(l_1, l_2)$ is computed by calculating the number of potential merges assuming that l_1 and l_2 are merged, that is, the number of potential merges in the set $\{(0, 4), (1, 3), (3, 4), (4, 5), (6, 4)\}$. In this case, there are four potential merges ($(0, 4)$ with $(4, 5)$, $(1, 3)$ with $(3, 4)$, $(3, 4)$ with $(4, 5)$, and $(6, 4)$ with $(4, 5)$) and thus, $weight(l_1, l_2) = 4$. Since $weight(l_1, l_2)$ is the largest, the heuristic will select to merge l_1 and l_2. After this merging the same process will be applied to the set of segments:

$$\{(l_{12} = (0, 4), l_3 = (1, 3), l_4 = (3, 4), l_5 = (4, 5), l_7 = (6, 4)\}$$

Notice that the segments $l_1 = (0, 2)$ and $l_2 = (2, 4)$ are replaced by their combined segment $l_{12} = (0, 4)$. l_3 and l_4 will be merged next, resulting the remaining set of $\{l_{12} = (0, 4), l_{34} = (1, 4), l_5 = (4, 5), l_7 = (6, 4)\}$. At last, l_{12} and l_5 will be merged and the algorithm terminates. Thus, for this example, the total number of shared ADMs is 5 and there are four segments $\{l_6, l_8\}$, $\{l_1, l_2, l_5\}$, $\{l_3, l_4\}$, and $\{l_7\}$. A potential wavelength assignment to achieve 5 shared ADMs is shown in Fig. 5.

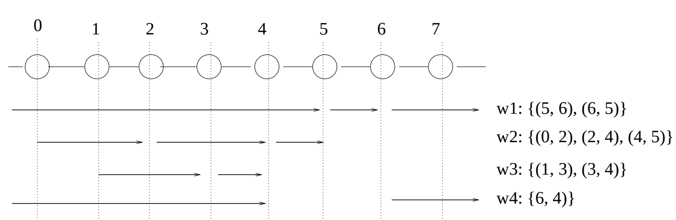


Figure 5: Final wavelength assignment for the example

6 Performance study

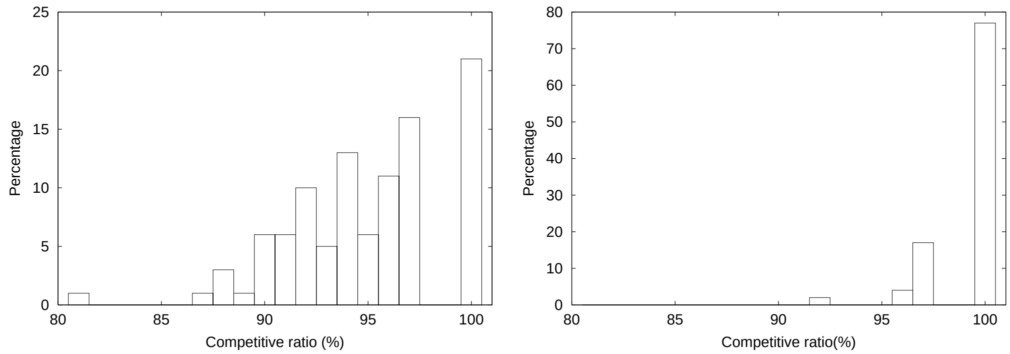
In this section, we evaluate the performance of the iterative merging heuristic, the best existing heuristic, and our newly proposed heuristic and compare their performance to the optimal solutions. The underlying ring network consists of 16 nodes (16 is recommended to be the maximal number of nodes for SONET rings).

Table 2 compare the performance of the heuristics with the optimal solutions. The optimal solutions are obtained by first reducing the problem size by finding all 2-lightpath circles and then solving the ILP formation for the rest of the lightpaths using lp_solve [7]. For a given number of lightpaths, we randomly generate all the lightpaths (random sources and random destinations) to form a wavelength assignment problem. The results reported in the table are the average of 100 randomly generated problems for each given number of lightpaths. The first column shows the number of lightpaths in the experiments. The second column shows the average of the maximum number of shared ADMs found using the ILP formation (optimal solutions). The third and the fourth columns show the performance of the iterative merging algorithm. The third column contains the number of shared ADMs found using the iterative merging algorithm. The fourth column shows the average competitive ratio of the iterative merging algorithm, that is, the ratio of the average number of shared ADMs found using the iterative merging algorithm and the average of the maximum number of shared ADMs. The fifth and the sixth columns show the performance of our heuristic. As shown in the table, both the iterative merging algorithm and our heuristic yield fairly good performance. In particular, for all these cases, the competitive ratio of our heuristic (versus the optimal solution) is more than 99%, which indicates that our heuristic is very effective.

Fig. 6 shows the distribution of the competitive ratio for the iterative merging heuristic and our heuristic. The statistic is obtained for 100 experiments with 70 lightpaths in a 16-node ring. As can be seen in the figure, our heuristic finds optimal solutions for about 77% of all cases. In the cases when our heuristic does not find optimal

No. of lightpaths	optimal solution (# of shared ADMs)	Iterative merging		Our algorithm	
		# of shared ADMs	ratio	# of shared ADMs	ratio
40	16.96	16.24	95.8%	16.88	99.5%
50	24.23	23.14	95.5%	24.02	99.1%
60	31.67	30.12	95.1%	31.46	99.3%
70	36.62	34.88	95.2%	36.36	99.3%
80	44.77	42.18	94.2%	44.37	99.1%

Table 2: Performance of the heuristics



(a) The iterative merging heuristic

(b) Our heuristic

Figure 6: The competitive ratio distribution

solutions, the results found by our heuristic is close to optimal. The iterative merging heuristic find the optimal solutions for only about 21% of all cases and its competitive ratio is widely distributed. This demonstrates that our heuristic is more effective and more robust than the iterative merging heuristic.

No. of lightpaths	upper bound (# of shared ADMs)	Iterative merging		Our algorithm	
		# of shared ADMs	ratio	# of shared ADMs	ratio
50	24.46	22.73	92.9%	23.64	96.6%
75	43.26	39.79	92.0%	41.64	96.3%
100	62.15	57.16	92.0%	59.47	95.7%
125	82.70	76.01	91.9%	79.04	95.6%
150	102.9	95.38	92.7%	98.54	95.7%

Table 3: Performance of the heuristics for larger problems

Table 3 shows the performance of the heuristics for larger problems. Due to the large problem size, the ILP formation cannot be solved efficiently and the optimal solution cannot be obtained. Here, we compute the upper bound for the maximum number of shared ADMs by relaxing the constraints to obtain integer solutions and transferring the problem into a linear programming problem that allows real solutions. This gives us an upper bound of the maximum number of shared ADMs. The results are the average of 100 randomly generated problems for each given number of lightpaths. The format of this table is the same as that of Table 2. The table shows that for larger problem sizes, the proposed heuristic also is very effective. The average number of shared ADMs found by the heuristic is within 95% of the upper bound.

7 Conclusion

When a physical optical WDM ring is used to support multiple SONET/SDH self-healing rings, 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 develop an integer linear programming formation for this problem, propose a new wavelength assignment heuristic, and evaluate the existing heuristic and our heuristic with the ILP formation. The results show that the newly proposed heuristic is more effective and more robust than the existing most effective heuristic, the iterative merging heuristic, and the performance of our heuristic is very close to that of the optimal algorithm.

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*, pages 1020-1025, 2000.

[7] LP_SOLVE, available at ftp://ftp.ics.ele.tue.nl/pub/lp_solve.

[8] Udi Manber, "Introduction to Algorithm: A Creative Approach", Addison-Wesley, 1989.