In *Proceedings of IASTED International Conference--Wireless and Optical Communications,*
July 17-19, 2002, Banff, Canada, pp. 333-338

Cost-Effective Approaches for Circuit Construction in WDM SONET Rings

Lin Li, Stephen Scott, and Jitender Deogun
Dept. of Computer Science and Engineering, University of Nebraska, Lincoln, NE 68588-0115, USA
email: {lili, sscott, deogun}@cse.unl.edu

Abstract

High speed SONET rings are widely used as network infrastructure. In WDM/SONET rings, the SONET add/drop multiplexers (SADMs) can significantly increase the overall system cost. By traffic grooming, the number of SADMs as well as the number of wavelengths required can be reduced dramatically. Generally, traffic grooming includes two parts: circuit construction and circuit grooming. Efficient circuit construction will help minimize not only the number of wavelengths but also the number of SADMs. In this paper, we focus on developing improved approaches for circuit construction under static traffic patterns, using hill climber and simulated annealing. Our algorithms show good results compared to previous work.

Keywords: SONET, WDM, SADM, traffic grooming, circuit construction, simulated annealing

1 Introduction

Much of today's network infrastructure is based on SONET rings. Because of the huge bandwidth of a wavelength, SONET allows each wavelength to carry multiple low-rate circuits simultaneously in a time division multiplexing (TDM) fashion. In the SONET rings, SONET add/drop multiplexers (SADMs) are used to multiplex multiple low-rate circuits (LRCs) into a high-rate ring for a certain wavelength. In WDM/SONET rings, if each node has to support every wavelength, it will need many SADMs and this will increase the electronics cost dramatically. Actually, not all the traffic must be electronically processed at each intermediate node, but only the wavelengths that carry message to and from that node. So optically bypassing some wavelengths of the intermediate nodes and efficiently assigning low-rate circuits, which is called *traffic grooming*, will lead to a reduction of electronics cost. Since the objective of network design is to minimize the overall system cost and the costs of the electronics dominate the network costs, the traffic grooming problem has become more and more important.

Some previous work, such as Gerstel et al. [2], considered lightpath assignment in SONET rings to reduce the number of SADMs. They proved that the number of wavelengths and SADMs could not be minimized concurrently, but they did not consider traffic grooming. Chiu & Modiano [4] proved that a general traffic grooming problem is NP-complete. Zhang & Qiao [3] extended previous work, discussed both unidirectional and bidirectional rings and provided a greedy approach. Wang et al. [5] gave a comprehensive mathematical definition of the general traffic grooming problem and improved the algorithm of Zhang & Qiao [3] using simulated annealing.

In this paper, we treat the traffic grooming problem as two separate parts as in Zhang & Qiao [3]. The first part is circuit construction, the second part is circuit grooming. Much work has been done on the second part, so we focus more on circuit construction. Under a static ring traffic pattern and assuming each node in the ring has one WADM and one or more SADMs, we consider efficient wavelength assignment in terms of low-rate circuit construction. Our objective is to efficiently construct the low-rate circuits before grooming them into the wavelengths, which would lead to a reduction of electronics cost. We employ hill climber and simulated annealing for optimization, and compare the results with previous work.

The rest of the paper is organized as follows: in Section 2, we describe the characteristics of circuit construction and the related work. In Section 3, we review the circuit construction problem in both uniform and non-uniform traffic patterns and then present our algorithms. In Section 4, numerical results are analyzed and compared. Finally, in Section 5, we give the conclusion of the paper and propose some future work.

2 Background and Related Work

2.1 Circuit Construction

Throughout the paper, we use the following notation:
N: *number of nodes in ring network*;
E: *number of end nodes in the network*;
C: *number of low-rate circuits a wavelength can carry*;
T: *input traffic matrix*.

Circuit construction on a ring network is packing nonoverlapping traffic connections into circuits according to the traffic demands. Efficient circuit construction means packing the connections with as few end nodes and circuits as possible. This will lead to fewer wavelengths and SADMs required. In some cases, we may not be able to minimize end nodes and circuits simultaneously. Depending on different system requirements, there is always some tradeoff between them. Generally, circuit construction will affect traffic grooming in two aspects. The first is that the circuit number directly decides the wavelength number. The more compact the circuits, the smaller the number of circuits constructed will be and this will lead to fewer wavelengths required. A simple example is as follows[1]: Suppose each wavelength is able to support four circuits (e.g OC-3 on an OC-12 ring), and the traffic amount between each connection is unit circuit. Given N=4 (from 1 to 4), and the traffic connection set $S = \{(1,2),(2,1),(1,3),(3,1),(2,3),(3,2),(2,4),(4,2)\}$, because every different node in the wavelength will require an SADM, six SADMs and two wavelengths are used in Assignment 1 and four SADMs and one wavelength are used in Assignment 2 by packing more circuits into one wavelength.

Assignment # 1	**Assignment # 2**
$\lambda_1: 1 \leftrightarrow 2,\ 1 \leftrightarrow 3$	$\lambda_1: 1 \leftrightarrow 2,\ 1 \leftrightarrow 3,\ 2 \leftrightarrow 3,\ 2 \leftrightarrow 4$
$\lambda_2: 2 \leftrightarrow 3,\ 2 \leftrightarrow 4$	

The second aspect is to minimize the number of different end nodes used in all connections in each wavelength by end node sharing. The end nodes include both the start and end nodes of a connection, which further helps reduce SADM number (note that end node number does not necessarily equal SADM number). An illustrative example is as follows: Suppose that each wavelength is able to support four circuits, and the traffic amount between each connection is also four circuits. Given N=5 (from 0 to 4) and traffic connection set $S = \{(0,1),\ (2,4),\ (1,3)\}$, by efficient circuit construction Assignment 2 saves one more SADM than Assignment 1 by maximizing end node sharing—without leaving a gap between nodes 1 and 2 in λ_1.

Assignment # 1	**Assignment # 2**
$\lambda_1: 0 \rightarrow 1,\ 2 \rightarrow 4$	$\lambda_1: 0 \rightarrow 1,\ 1 \rightarrow 3$
$\lambda_2: 1 \rightarrow 3$	$\lambda_2: 2 \rightarrow 4$

A general traffic grooming problem involves both of these parts. In practice, we always try to pack a wavelength with as many circuits as possible to save wavelengths and we always try to maximize end node sharing. Circuit construction is the basic step of traffic grooming. Once it is finished efficiently, wavelength assignment can be largely determined.

2.2 Related Work and Our Contribution

Gerstel et al. [2] proposed two heuristics for the lightpath assignment problem, which is actually a special case of circuit construction. One is **Cut-first**, and the other is **Assign-first**. The underlying assumption was that the calls between every node pair required a full wavelength and no grooming was considered. In cut-first, the algorithm first finds a node which may bring the most possible end node sharing. Then all connections across this node will be split and the updated traffic matrix will be assigned according to some principle. The difference between cut-first and assign-first is that the latter never splits the existing traffic connections but tries to assign them as efficiently as possible. Gerstel et al. [2] also addressed some possible post-assignment transformation operators: *merging, combining,* and *splitting*. Merging refers to swapping two valid segments with the same start and end nodes in two different wavelengths to see if this will bring more end node sharing. A valid segment is a segment in a wavelength that includes integral (i.e. whole) connections and partial or integral gaps. A simple example is in Figure 1.

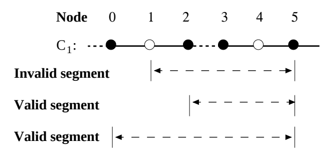

Figure 1. Examples of valid and invalid segments.

Combining refers to finding two segments of unnecessarily split connections and combining them. Splitting refers to splitting a connection such that the split parts can fit perfectly into the gaps of other wavelengths. All these operators may bring more end node sharing and make the assignment more efficient. Wan et al. [8] proposed various greedy approaches based on circular arc grouping and pointed out that these approaches could be applied to circuit construction. Later, Liu et al. [9] gave an approximation algorithm for the NP-hard wavelength assignment problem and also gave preliminary experimental results using the proposed greedy heuristics of Wan et al. [8]. In contrast, our work is based on simulated annealing and random hill climber, which in practice have been observed to be better than greedy approaches in avoiding local optima. Our algorithms are also more aggressive in splitting connections to achieve more efficient circuit construction, as described below.

Though cut-first and assign-first were proposed for lightpath assignment, we found the algorithms very useful in circuit construction. In fact, if we replace the whole wavelength in cut-first and assign-first with a single circuit, then by splitting, combing, and merging operations, we may construct the circuits more efficiently, which will lead to much fewer circuits and end nodes. So we adopted these two algorithms as the first step and used hill climber and simulated annealing [7] to develop our algorithms. Moreover, we give a comprehensive integer cubic program model to describe an optimal solution to this problem.

[1] $i \rightarrow j$ represents one traffic connection from i to j; $i \leftrightarrow j$ represents two connections: $i \leftarrow j$ and $j \leftarrow i$, which form a circle.

3 Traffic Pattern and Our Algorithms

3.1 Uniform Traffic in Unidirectional Rings

In the uniform traffic model, the traffic is constant for any pair of nodes, denoted by b. And each node has some message to send to every other node. If the traffic amount between every node pair is a unit circuit, $(b = 1)$, we can establish a total of $\frac{N(N-1)}{2}$ full circles (i.e. every circle is composed of two connections, one is from node i to j, the other is from node j to node i). The capacity of each circle equals the number of low-rate circuits required between any node pair. Correspondingly, there will be $\frac{N(N-1)b}{2}$ low rate circuits when $b \geq 1$. Note that since having full circles will lead to a reduction of SADMs [3], we always try to construct as many full circles as possible. Depending on whether the wavelength capacity is a multiple times of the traffic amount between every node pair, we can determine whether to groom the b circles into more than one wavelength or not. Some related algorithms have been proposed as in [3, 4].

Circuit construction under uniform traffic is easy to implement. So efficient circuit grooming is the key problem, i.e. how to select C low-rate circuits to minimize end node sharing when packing them into a certain wavelength. One principle is always selecting a circuit that has as many overlapping end nodes as possible. But this may trap into some locally optimal result. An illustrative example is: in a ring with 6 nodes (N=6), one circuit traffic amount between every pair of nodes, three circuits per wavelength (e.g OC-48 on an OC-192 ring). Under the uniform traffic model, we have $\frac{N(N-1)}{2} = 15$ circuits, which can be groomed into 5 wavelengths, and we try to minimize the total SADMs required.

The 15 circuits corresponding to 15 different node pairs are:

(1,2) (2,3) (3,4) (4,5) (5,6)
(1,3) (2,4) (3,5) (4,6)
(1,4) (2,5) (3,6)
(1,5) (2,6)
(1,6).

If we try to overlap end nodes as much as possible when grooming circuits to each wavelength, we may get Assignment 1 with 18 SADMs required. But Assignment 2 only

Assignment # 1	**Assignment # 2**
λ_1: $1 \leftrightarrow 2$, $1 \leftrightarrow 3$, $2 \leftrightarrow 3$	λ_1: $1 \leftrightarrow 2$, $1 \leftrightarrow 3$, $2 \leftrightarrow 3$
λ_2: $1 \leftrightarrow 4$, $1 \leftrightarrow 5$, $4 \leftrightarrow 5$	λ_2: $1 \leftrightarrow 4$, $1 \leftrightarrow 5$, $4 \leftrightarrow 5$
λ_3: $2 \leftrightarrow 4$, $2 \leftrightarrow 6$, $4 \leftrightarrow 6$	λ_3: $2 \leftrightarrow 4$, $2 \leftrightarrow 6$, $4 \leftrightarrow 6$
λ_4: $3 \leftrightarrow 5$, $3 \leftrightarrow 6$, $5 \leftrightarrow 6$	λ_4: $3 \leftrightarrow 4$, $3 \leftrightarrow 5$, $3 \leftrightarrow 6$
λ_5: $1 \leftrightarrow 6$, $2 \leftrightarrow 5$, $3 \leftrightarrow 4$	λ_5: $1 \leftrightarrow 6$, $2 \leftrightarrow 5$, $5 \leftrightarrow 6$

requires 17 SADMs. Even though in Assignment 1, the first five wavelengths all have the maximum overlapping end nodes, and only three in assignment 2, the latter leads to a globally optimal result.

3.2 Non-uniform Traffic in Unidirectional Rings

The uniform traffic model is not very practical. Studying the characteristics of non-uniform traffic is more important. In non-uniform traffic, it is not always possible to construct full circles since the traffic amount between different node pairs is different. So circuits with gaps are introduced.

Based on the previous discussion, since a partial circuit with gaps will waste bandwidth, we still try to construct as many full circles as possible. But in non-uniform traffic, how can we select different connections to construct low-rate circuits efficiently? Zhang & Qiao [3] proposed an algorithm that assigns the longest connection first, then fits the shorter connections into the gaps. The idea is to expect that the shorter connections are more likely to be able to fit into the gaps generated by the longer connections. Actually, Zhang & Qiao [3] and many others' work mainly focused on how to assign the existing connections in the traffic matrix by employing greedy approaches. The connections in the traffic matrix are never changed. However, as we discussed in Section 2.2, when a traffic connection is split into smaller parts, more end node sharing may be achieved. An illustrative example is as follows. Given N=10, C=3 and a non-uniform traffic connection set $S = \{(0,6),(6,2),(2,8),(8,4),(4,0)\}$,

Assignment # 1	**Assignment # 2**
C_1: $0 \rightarrow 6$	C_1: $0 \rightarrow 6$, $6 \rightarrow 0$
C_2: $6 \rightarrow 2$	C_2: $0 \rightarrow 2$, $2 \rightarrow 8$, $8 \rightarrow 0$
C_3: $2 \rightarrow 8$	C_3: $0 \rightarrow 4$, $4 \rightarrow 0$
C_4: $8 \rightarrow 4$	
C_5: $4 \rightarrow 0$	

since each connection traverses 6 links (more than halfway around the ring) any two connections will overlap each other if they are assigned into the same circuit. So each connection has to be assigned to different circuits without splitting. A total of five circuits are used as in Assignment 1. Given N=10 and C=3, at least two wavelengths are required to accommodate all these circuits and at least seven SADMs are required. But if we take all connections crossing node 0, and split each into two parts with node 0 as the break point, the new traffic set would be $S' = \{(0,6),(6,0),(0,2),(2,8),(8,0),(0,4),(4,0)\}$. Correspondingly, we can get another assignment of the set as in Assignment 2. Now the resulting connections can fit perfectly into 3 circuits. Given N=10 and C=3, one wavelength has enough capacity to accommodate all these circuits and five SADMs are required. Compared to Assignment 1, two SADMs are saved. Note that splitting the connections will lead to more connections in the traffic matrix, but more shorter connections are easier for us to construct full low-rate circuits, which further leads to a decrease of wavelength number. Moreover, we can also increase the probability of end node sharing and this will lead to more SADM saving.

Assigning the traffic matrix without any connection change is called *single-hop assignment* since all connections are direct connections. Assigning the traffic matrix after some connection splitting is called *multihop assignment* since some intermediate nodes are introduced for bridging the connections. From Assignment 2, we can see that 5 connections need to stop at and begin from node 0. Generally, connection splitting will increase the signal processing burden of the nodes and delay the signal transmission. This brings a performance evaluation problem, which we will not discuss in this paper. Some previous work discussed multihop traffic grooming based on hub structure [6, 10]. The differences between our work and theirs are: first, no specific node is assigned as a hub. By iterative splitting and combining unnecessary splitting, we distributed the bridging function of a hub to several nodes. So there is no requirement for some nodes equipped with an expensive digital crossconnect or SADMs for each wavelength. Second, utilizing different routing schemes, all connections are established on the shortest path and our algorithms always construct fewer low-rate circuitis than algorithms that assign the traffic without connection changes. But algorithms based on hub structure do not have this property.

3.3 Integer Cubic Program Formulation

The circuit construction problem can be mathematically defined as in Figure 2, where i identifies a connection, j identifies a node, and k identifies a circuit. A solution to the circuit construction problem is represented by variables E_{ijk} and I_{ijk} for $1 \leq i \leq N^2G$ (G is the maximum entry of the traffic matrix), $1 \leq j \leq N$, and $1 \leq k \leq H$ (H is the input circuit number). $E_{ijk} = 1$ if in the solution, node j in circuit k is an end node in connection i, and 0 otherwise. Likewise, $I_{ijk} = 1$ if in the solution, node j in circuit k is an intermediate node in connection i, and 0 otherwise (note that since a connection can be split in a solution, it could have several E_{ijk} variables equal to 1 for a given i). B_{jk} represents the end nodes in circuit k at position j. B_{jk} is 1 if node j in circuit k is an end node for any connection (or pair of connections), and 0 otherwise. This variable counts the total number of end nodes used in a solution, and thus is what is minimized in the objective function. M is the number of total connections in a given traffic matrix. An illustrative example is shown in Figure 3. In the circuit assignment after splitting, connection l is split into two connections, connection l_1 and connection l_2, which require two more end nodes—E_{l3c_1} and E_{l3c_2}. Since node 2 is shared by connection i and connection l, E_{i2c_1} and E_{l2c_1} are both 1, but we only want to count that node once as end node. Thus to ensure the end node is counted only once, we count B_{2c_1} in the objective function. Finally, we let st_i equal the index of the start node of connection i (from the traffic matrix), and end_i the index of end node of connection i. $Length_i$ represents the length of connection i, which equals $(st_i - end_i + N) \pmod N$.

Objective function:

$$Minimize: \sum_j \sum_k B_{jk}$$

Subject to:

1. each node is either an end node or an intermediate node:

$$E_{ijk} + I_{ijk} \leq 1 \qquad \forall i, j, k$$

2. each connection is completely assigned:

$$\sum_k \sum_j (E_{ijk} + I_{ijk}) - \frac{1}{2} \sum_k \sum_j E_{ijk} = Length_i \quad \forall i$$

3. no two connections share a link:

$$\sum_i (E_{ijk} + 2I_{ijk}) \leq 2 \qquad \forall j, k$$

4. no segments from the same connection overlap:

$$\sum_k (E_{ijk} + 2I_{ijk}) \leq 2 \qquad \forall i, j$$

5. the intermediate nodes must be between end nodes:

$$\sum_k \sum_{\substack{j_s, j_e \in \{st_i, \ldots, end_i\} \\ j_s < j_e}} (E_{ij_sk} + E_{ij_ek} + \sum_{j'=j_s+1}^{j_e-1} (E_{ij_sk} I_{ij'k} E_{ij_ek}))$$

$$-(Length_i - \frac{1}{2}) \sum_k \sum_j E_{ijk} = Length_i \quad \forall i$$

5′. all nodes between end nodes are marked as intermediate:

$$\sum_k \sum_{\substack{j_s, j_e \in \{st_i, \ldots, end_i\} \\ j_s < j_e}} (E_{ij_sk} + E_{ij_ek} + \sum_{j'=j_s+1}^{j_e-1} (E_{ij_sk} E_{ij_ek}))$$

$$-(Length_i - \frac{1}{2}) \sum_k \sum_j E_{ijk} = Length_i \quad \forall i$$

6. each end node is counted only once:

$$\sum_i E_{ijk} \leq B_{jk} M \qquad \forall j, k$$

Bounds:

All E_{ijk}, I_{ijk}, and $B_{jk} \in$ {0, 1}.

Figure 2. The integer cubic program model

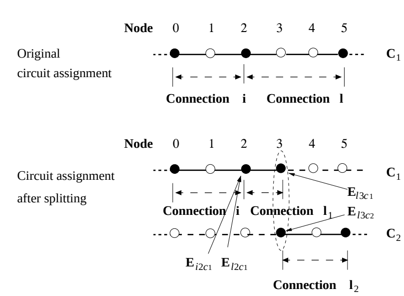

Figure 3. An example of the variables in Figure 2

The first constraint states that every node in the circuits can be an end node, an intermediate node, or neither. The second constraint states every connection in the traffic matrix has to be completely assigned. The third constraint states no two connections can share the same link in a circuit. The fourth constraint states no two segments from the same connection overlap. The fifth constraint states the intermediate nodes of a connection can only lie between its end nodes and all nodes between end nodes are marked as intermediate nodes. The last constraint is to ensure an end node is only counted once. Note that in constraint 4, we require that a solution be assigned such that for each connection i, any LRC k is used at most once, i.e., for any valid solution s such that k is used 2 or more times in the same i can be mapped to another valid solution s' without this case where s' has an equal or smaller end node cost than s. A detailed proof of this is deferred to the full version of this paper.

3.4 Random Hill-Climber and Simulated Annealing

We propose applying the Random Hill-climber and Simulated Annealing strategies to solve the circuit construction problem. The output of applying cut-first heuristic is adopted as the starting point for our approaches. We start with the output of cut-first because cut-first always generates several short connections, which will lead to fewer circuits constructed. Moreover, applying the merging, splitting, and combining, operators of Section 2.2 is closely related to cut-first. The hill climber algorithm is hereafter referred as **Algorithm I** and Simulated Annealing algorithm is as **Algorithm II**.

The basic principle of **Algorithm I** is: from a current solution, a set of new solutions is generated by applying the operators. The new current solution is then selected randomly from the solutions in this set that are better than the current solution. In this way, we can reach a good solution along a certain path. After storing a number of paths, the best overall solution is returned.

In **Algorithm II**, we apply the same operators as in **Algorithm I**, but we use a different strategy to select new solutions. Specifically, we choose a random solution s from current solution s_1 and make it the current solution with probability $\min\{1, e^{-\delta/T}\}$, where $\delta = cost1 - cost2$, $cost1$ is the number of end nodes of solution s, and $cost2$ is the number of end nodes of solution s_1. T is the "temperature" parameter of simultad annealing. In this way, we may escape getting trapped in local optima. As in **Algorithm I**, **Algorithm II** returns the best solution found over all paths.

4 Empirical Results

For experimental evaluation, we generated non-uniform traffic matrices T in a ring with N nodes. Each entry t_{ij} in T represents the traffic amount in units of LRCs from node i to node j. We calculate the efficiency of the algorithms by by randomly generating 20 traffic matrices for each value of $N \in \{5, 6, 7, 8, 9, 10\}$ and averaging the final results. To simulate non-uniform traffic characteristics, each t_{ij} is a random number between 0 and 10 ($t_{ii} = 0$).

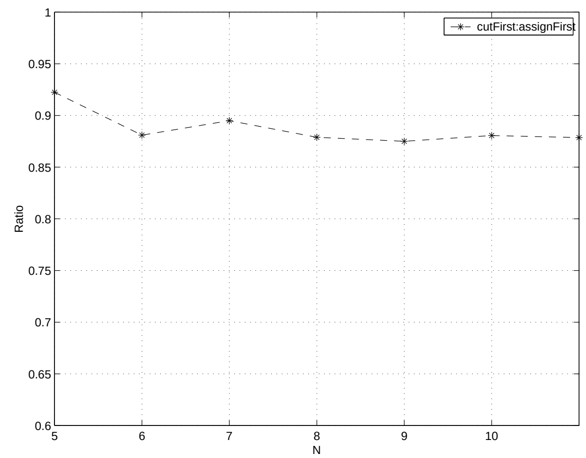

Figure 4. Ratio of LRC number of cutFirst to assignFirst

Figure 4 depicts the ratio of LRC number of cut-first to that of assign-first. Note that the splitting and reassigning of connections done by cut-First leads to a solution that uses about 10% fewer circuits than those used by solutions produced by assign-First. Thus since our algorithms start with the output of cut-First, they also enjoy this savings of LRCs.

Figures 5 and 6 depict the ratio of end node number after applying the algorithms. From the curves we can see that **Algorithm I** and **Algorithm II** both produce solutions with nearly 10% fewer end nodes than cut-first and assign-first. This will greatly help reduce the number of SADMs and time complexity when applying circuit grooming algorithms. Another interesting thing is it seems hill climber is more stable than simulated annealing. **Algorithm I** always shows good performance in the experiments. While simulated annealing may win over hill climber when N is small, with increasing N, it is hard for **Algorithm II** to achieve the same good results.

Note that since **Algorithm I** and **Algorithm II** are based on using fewer circuits first, there is a chance that assign-first algorithm will result in fewer end nodes while using more circuits. But from the curves, the average overall performance is greatly improved.

5 Conclusions and Future Work

Through improved approaches, we can construct low-rate circuits in SONET rings more efficiently. Both circuit number and end node number are reduced currently, which will lead to a reduction of wavelength number and SADM number. Though end node number does not equal SADM number, a solution with a reduced number of end nodes is likely

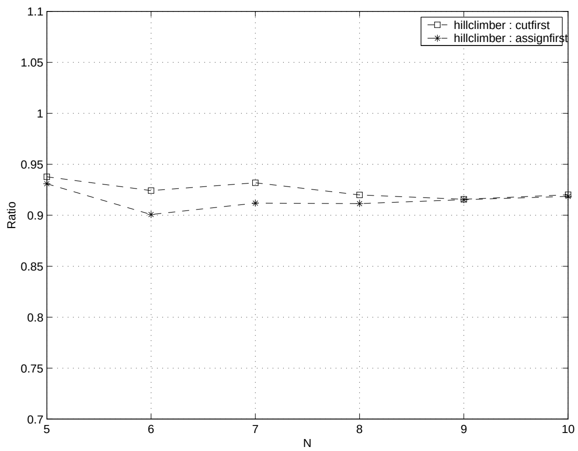

Figure 5. Ratio of end node number of hillclimber to cutFirst and assignFirst

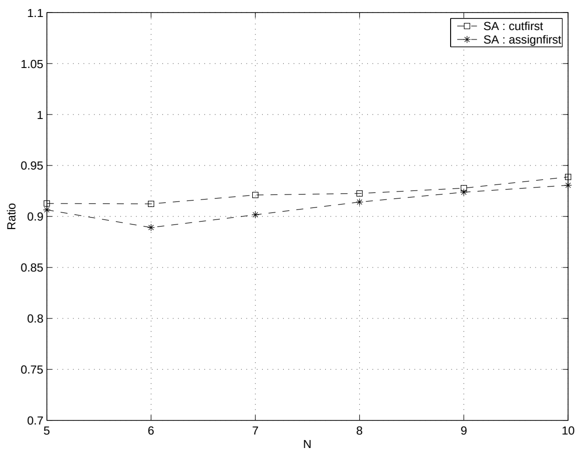

Figure 6. Ratio of end node number of simulated annealing to cutFirst and assignFirst

to require fewer SADMs. Such a solution also reduces complexity when applying circuit grooming algorithms.

Plans for future work include bidirectional circuit construction, measuring the final result after applying kinds of circuit grooming algorithms [3][5], and evaluating the integer cubic program in Section 3.4 to lower bound the curves in Figures 4-6.

Acknowledgments

This work was funded in part by NSF grants CCR-9877080 (with matching funds from UNL-CCIS) and CCR-0092761.

References

[1] E. Modiano, and P. Lin, “Traffic grooming in WDM networks”, *IEEE communication magazine*, vol. 39, pp. 124-129, 2001.

[2] O. Gerstel, P. Lin, and G. Sasaki, “Wavelength Assignment in a WDM ring to minimize cost of Embedded SONET Rings”, *Proc., IEEE INFOCOM’98*, pp. 94-101, March 1998.

[3] X. Zhang and C. Qiao, “An effective and comprehensive solution for traffic grooming and wavelength assignment in SONET/WDM rings”, *IEEE/ACM Transactions on Networking*, Vol. 8, pp. 608-617, 2000.

[4] A. Chiu, and E. Modiano, “Traffic grooming algorithms for reducing Electronic Multiplexing Costs in WDM ring Networks”, *IEEE/OSA Journal of lightwave technology*, vol. 18, pp. 2-12, 2000.

[5] J. Wang, V.Rao Vemuri, W. Cho, and B. Mukherjee, “Improved Approaches for Cost-Effective Traffic Grooming in WDM Ring Networks: Nonuniform Traffic and Bidirectional Ring”, *ICC2000*, pp. 1295-1299, 2000

[6] J. Wang, W. Cho, V. Rao Vemuri, and B. Mukherjee, “Improved Approaches for Cost-Effective Traffic Grooming in WDM Ring Networks: ILP Formulations and Single-Hop and Multihop Connections”, *IEEE/OSA Journal of lightwave technology*, vol. 19, pp. 1645-1653, Nov, 2001.

[7] Z. Michalewicz, and D.B.Fogel, *How to Solve it–Modern Heuristics*, Springer 2000.

[8] P. Wan, L. Liu, and O. Frieder, “Grooming of Arbitrary Traffic in SONET/WDM rings”, *Proc., Globalcom’99*, pp. 1012-1016, 1999.

[9] L. Liu, X, Li, P. Wan and O. Frieder, “Wavelength Assignment in WDM Rings to Minimize SONET ADMs”, *Proc., IEEE INFOCOM’00*, pp. 1020-1025, 2000.

[10] O. Gerstel, R. Ramaswami, and G. Sasaki, “Cost Effective Traffic Grooming in WDM Rings”, *IEEE/ACM Transactions on Networking*, vol. 8, pp. 618-30, 2000.