2002

Dynamic traffic grooming algorithms for reconfigurable SONET over WDM networks

Shu Zhang
University of Nebraska - Lincoln, shuz@cse.unl.edu

Byrav Ramamurthy
University of Nebraska - Lincoln, bramamurthy2@unl.edu

GLOBECOM '02. IEEE Global Telecommunications Conference, 2002.
Volume: 3 Digital Object Identifier: 10.1109/GLOCOM.2002.1189123
Publication Year: 2002 , Page(s): 2716 - 2720 vol.3

Dynamic traffic grooming algorithms for reconfigurable SONET over WDM networks

Shu Zhang and Byrav Ramamurthy
Dept.of Computer Science and Engineering, University of Nebraska-Lincoln, USA
E-mail: {shuz, byrav}@cse.unl.edu

***Abstract*—The emergence of Wavelength Division Multiplexing (WDM) technology provides the capability for increasing the bandwidth of Synchronous Optical Network (SONET) rings by grooming low-speed traffic streams onto different high-speed wavelength channels. Since the cost of SONET add-drop multiplexers (SADM) at each node dominates the total cost of these networks, how to assign the wavelength, groom the traffic and bypass the traffic through the intermediate nodes has received a lot of attention from researchers recently. Moreover, the traffic pattern of the optical network changes from time to time. How to develop dynamic reconfiguration algorithms for traffic grooming is an important issue. In this paper, two cases (best-fit and full-fit) for handling reconfigurable SONET over WDM networks are proposed. For each approach, an integer linear programming model and heuristic algorithms (based on the tabu search method) are given. The results demonstrate that the tabu search heuristic can yield better solutions but has a greater running time than the greedy algorithm for the best-fit case. For the full-fit case, the tabu search heuristic yields competitive results compared with an earlier simulated annealing based method and it is more stable for the dynamic case.**

I. Introduction

Most of today's optical networks are built on SONET rings [1]. Using WDM technology, multiple rings can be supported on a single fiber ring [3]. In this architecture, each wavelength independently carries a SONET ring. Each SONET ring can further support multiple low speed streams (e.g. an OC-48 SONET ring can support 4 OC-12 or 16 OC-3 streams at the same time). At each node a WDM Add/Drop Multiplexer (WADM) adds and drops or bypasses traffic on any wavelength. At each node, there are SONET add/drop multiplexers (SADM) on each wavelength to add/drop low speed streams. So the number of SADMs per node will increase linearly with the number of wavelengths that a single fiber ring can carry. The cost of SADMs will dominate the total cost of the optical network. But in fact, it is not necessary for each node to be equipped with SADMs on each wavelength. The problem of combining different low speed traffic streams into high-speed traffic streams in such a way that the number of SADMs is minimized is called traffic grooming. Several studies have been done on traffic grooming [2], [5], [6].

However, most algorithms assume the traffic matrix to be static; actually, the traffic pattern over SONET rings changes from time to time. In [7], dynamic traffic is described by a multiple set of the traffic matrices and traffic grooming solution is proposed to meet the multiset instead of a single matrix. However, it is common that a change of traffic matrix happens after the configuration is established. In this paper, we consider the dynamic traffic grooming problem incorporating reconfiguration. That is, based on the current wavelength assignment, when the traffic pattern of the networks changes, we propose a dynamic traffic grooming algorithm to reconfigure the wavelength assignment according to the new traffic pattern without disrupting the old traffic assignment. Two cases, Best-fit and Full-fit are studied. Two heuristic algorithms, Greedy and Tabu-search (TS-1) are presented for the Best-fit approach. A two phase algorithm based on tabu search (TS-2) is presented for the Full-fit approach. The static traffic grooming problem, on which many studies have been done so far, is a special case of the *dynamic* problem (specially, the full-fit case). The simulation results illustrate that the tabu search algorithm can yield better solutions but takes more running time than greedy algorithms for the best-fit case. The tabu-search algorithm for the full-fit case is more stable compared to the earlier simulated annealing heuristic [5]. Some results for the static grooming problem are found to be better than earlier results in [5].

II. Problem Definition: Dynamic Traffic Grooming

In this section, integer linear programming models are proposed for dynamic traffic grooming. They are based on the static models proposed in [5]. There are two assumptions.

1. The old traffic matrix is known. The current configuration of the network is obtained according to the old matrix. This corresponds to the assignment of traffic to wavelengths in the network.

2. The traffic matrix changes. The new matrix is different from the original matrix. The objective is to disrupt as few current connections as possible and fit the new traffic requests in.

The following are the notations that we will use later.

1. There are N nodes numbered $0, 1, 2, \ldots N-1$ in the SONET ring.

2. W is the number of wavelengths in the original traffic grooming matrix.

3. The granularity of the traffic g is defined as:

$$g = \frac{channel\ capacity}{base\ bandwidth\ rate}$$

e.g. for a OC-48 channel carrying several OC-3 streams, the granularity $g = \frac{OC-48}{OC-3} = 16$. It is the number of circles [1] C each wavelength can carry.

4. Original traffic matrix $T[n \times n]$.

The traffic amount from node i to node j ($i, j = 1..N-1$, $i \neq j$) on the ring is denoted by t_{ij} (entry of matrix T at row i and column j) and is always a multiple of the base bandwidth rate.

5. $T'[n \times n]$ is the new traffic matrix.

6. V_{ij}^{cw} represents the number of virtual connections for node pair (i, j) on wavelength w in circle c according to the original matrix. It is known in the reconfiguration problem.

This work was supported by NSF grants (ANI-0074121 and EPS-0091900).

[1] The circle here refers to the circle built by the algorithm in [6].

0-7803-7632-3/02/$17.00 ©2002 IEEE

7. $V_{ij}^{'cw}$ represents the number of new virtual connections from i to j on wavelength w of circle c according to the new traffic matrix.
8. V_{ij}^{cw+} represents the number of new virtual connections from i to j on new wavelength w^+ of circle c according to the new traffic matrix (applies to full-fit case only).
9. ADM_i^w represents the number of ADMs on node i on wavelength w for the original matrix.
10. $ADM_i^{'w}$ represents the number of additional ADMs on node i on wavelength w (applies to full-fit case only).
11. ADM_i^{w+} represents the number of ADMs on node i on new wavelength w^+ (applies to full-fit case only).

We assume that for the original traffic matrix, a solution has been found (e.g. using the algorithm in [6], [5]). So we know the number of wavelengths W and the virtual connections for each pair (i, j). In addition, we also know the the number of SADMs and their positions.
Without special indication, unidirectional rings are assumed in the following problem descriptions. For bi-directional rings, constraints for both directions should be satisfied. Here are two cases to consider in the problem:

A. Best-fit case

Without increasing the number of SADMs, we try to place as much new traffic as possible.
$Maximize : \sum_i \sum_j \sum_c \sum_w V_{ij}^{'cw}$
(Objective function).
The objective is to maximize the traffic amount according to the new traffic demand. The following constraints are assumed.
1. Traffic constraint:
$\sum_w \sum_c V_{ij}^{'cw} + \sum_w \sum_c V_{ij}^{cw} \leq t_{ij}^{'}$
The traffic constraint indicates that the total number of virtual connections from i to j should be less than the new demand traffic matrix of entry t'_{ij}.
2. Circle capacity constraint:
$\sum_{(i,j)e\in(i,j)} (V_{ij}^{'cw} + V_{ij}^{cw}) \leq 1$
The circle capacity constraint requires that no two connections can share a link on a circle.
3. Transmitter constraint:
$\sum_c \sum_j V_{ij}^{'cw} + \sum_c \sum_j V_{ij}^{cw} \leq g \cdot ADM_i^w$
The transmitter constraint requires that the total number of virtual connections starting at node i should be less than the transmitter capacity of SADM at this node on wavelength w.
4. Receiver constraint:
$\sum_c \sum_i V_{ij}^{'cw} + \sum_c \sum_i V_{ij}^{cw} \leq g \cdot ADM_j^w$
The receiver constraint requires that the total number of virtual connections terminating at node j should be less than the receiver capacity of SADMs at this node on wavelength w.
$V_{ij}^{cw'} \in \{0, 1, -1\}$. $ADM_j^w \in \{0, 1\}$. $V_{ij}^{cw'}$ = -1 only if V_{ij}^{cw} = 1 and there is no connection between i and j on wavelength w of circle c for the new configuration any more.

B. Full-fit case

Add the minimum number of SADM to satisfy all the new traffic.
$Minimize : \sum_i \sum_w ADM_i^{'w} + \delta \sum_i \sum_{w+} ADM_i^{w+}$ (Objective function).
The objective is to fit all the traffic with the minimum number of SADMs added. δ in the objective function is the weight parameter representing the cost of adding more wavelengths. Because usually adding new wavelengths will cost more than adding SADMs on existing wavelengths, δ is supposed to be no less than one. The following constraints are assumed.
1. Traffic constraint:
$\sum_w \sum_c (V_{ij}^{cw} + V_{ij}^{'cw}) + \sum_{w+} \sum_c V_{ij}^{cw+} = t'_{ij}$
The traffic constraint indicates that the total number of virtual connections from node i to node j should equal the traffic demand from node i to j.
2. Circle capacity constraint:
$\sum_{(i,j)e\in(i,j)} (V_{ij}^{cw} + V_{ij}^{'cw}) \leq 1$
$\sum_{(i,j)e\in(i,j)} V_{ij}^{cw+} \leq 1$
The circle capacity constraint requires that no two connections can share a single link on a circle.
3. Transmitter constraint:
$\sum_c \sum_j (V_{ij}^{cw} + V_{ij}^{'cw}) \leq g \cdot (ADM_i^w + ADM_j^{'w})$
$\sum_c \sum_j V_{ij}^{cw+} \leq g \cdot ADM_i^{w+}$
The transmitter constraint requires that the total number of virtual connections should be less than the transmission capacity of the equipment at this node.
4. Receiver constraint:
$\sum_c \sum_i (V_{ij}^{cw} + V_{ij}^{'cw}) \leq g \cdot (ADM_j^w + ADM_j^{'w})$
$\sum_c \sum_i V_{ij}^{cw+} \leq g \cdot ADM_j^{w+}$
The receiver constraint requires that the total number of virtual connections should be less than the receiving capacity of the equipment at this node.
$V_{ij}^{'cw} \in \{0, 1, -1\}$. V_{ij}^{cw+}, $ADM_j^{'w}$ and $ADM_j^{w+} \in \{0, 1\}$.

As we know, the integer linear programming problem is NP-complete [4]. The reconfiguration problem is described based on integer linear programming models. We expect this problem also to be intractable. In the next section, we propose heuristic approaches to solve this problem.

III. Heuristic algorithms

The heuristic algorithms for dynamic grooming were developed for both the best-fit case and the full-fit case.

A. Best-fit

The objective of the best-fit case is to include as much new traffic as possible using available capacity of the current configuration of the ring networks without increasing the number of SADMs.

A.1 Greedy heuristic

In our greedy algorithm, the value of each entry of both the old matrix from which the current configuration was obtained and the new matrix using which we will perform the reconfiguration is generated randomly in the range $[0, r]$. The value is uniformly distributed between 0 and r. We try to fit as much new traffic as possible without adding SADMs. Here is a description of the algorithm.

1. Get grooming information for the original traffic matrix using an existing algorithm (e.g., [6], [5]).
2. Find the difference traffic matrix. Given the new traffic matrix, compute the difference between the old one and the new one e.g. $D[i, j] = T'[i, j] - T[i, j]$. This is the matrix we try to groom in our algorithm with existing SADMs and traffic capacity.

For some entries $D[i,j] = -m < 0$, (there exist some connections built for the old matrix that are not needed in new matrix any more), remove the connection between (i,j) from m circles over at most m wavelengths.

3. Merge connections. We want to keep the largest continuous gap between nodes. So if there are two continuous connections over two circles, we merge them into a bigger one on the same circle.

4. Groom new traffic. Start from the smallest hop. Groom traffic into the circle when there is enough capacity and there are SADMs on the terminating node.

A.2 Tabu search heuristic (TS-1)

Tabu search is a meta-heuristic approach to solve hard optimization problems. The optimizing function is $f(x)$ subject to $x \in X$. The set X summarizes constraints on the vector of decision variables x. if x is the initial solution, neighborhood $N(x)$ is a set obtained by going one step further from the solution x. Such a step is called a move. Each element in N(x) is put into the candidate list. At the same time, a tabu list is built to keep track of the solutions that have been visited before. Each element in the tabu list has a tabu tenure. After each move, the value of tabu tenure is decremented by one. Once the value becomes zero, the element is removed from the list. The best solution in the candidate list but not tabu (not in the tabu list) is chosen for the initial solution of next iteration. If after certain number of iterations (we call it the tabu limit) which is specified by the user, there is no improvement, the program stops. Otherwise, we continue the iteration and build a new candidate list.

In TS-1, the initial solution is the solution obtained from the greedy heuristic proposed in previous section. The neighborhood $N(x)$ is defined as: $N(x) = \{x' \mid x'$ is a move by swapping part of the two circles of the solution x and one part of the swap should have available capacity $\}$. The non-tabu solution with the most number of new connections established is chosen for the next iteration. The tabu tenure for TS-1 is 48 and the tabu limit is 60.

B. Full-fit

A two phase algorithm is developed for the full-fit case. The old traffic matrix and its solution are known. Recall that each entry of the old matrix and new matrix is generated randomly ranging from $[0, r]$. The objective is to fit all new traffic requests. Adding SADMs is allowed.

B.1 Algorithm description

1. Use a best-fit algorithm (greedy or TS-1) to groom as much traffic as is allowed with existing SADMs.

2. If the capacity on circle c is available for the connection (i,j), place SADMs at nodes i and j on the wavelength which circle c is groomed on and groom traffic onto this circle.

3. Use the tabu search algorithm (TS-2) to groom the remaining traffic onto the new wavelength. Place SADMs at the nodes whenever necessary.

B.2 Tabu search heuristic (TS-2)

The problem that was solved by TS-2 is the static traffic grooming problem, on which a lot of work has been done so far [6], [5], [2], [8]. That is, given the remaining traffic matrix, groom that traffic onto wavelengths so that the number of SADMs is minimized. We observe that the static traffic grooming problem is a special case of the dynamic problem described in Section II-B when the old traffic matrix is empty and the new matrix is the traffic matrix that will be groomed. The initial solution x is obtained by using algorithm in [5], [6]. The neighborhood $N(x)$ is defined as: $N(x) = \{x' \mid x'$ is a move by swapping two circles from different wavelengths of solution $x\}$. The non-tabu solution with the minimum number of SADM is chosen from the candidate list for the next iteration. The tabu limit is 170 and tabu tenure is in $\{15, 20, 24\}$.

IV. Simulation Results and Analysis

In this section, we present the simulation results for both the best-fit case and the full-fit case.

A. Results for the best-fit case

First, we define an upper bound on the number of new connections that can be groomed to the current configuration to evaluate the performance of the best-fit case algorithms. Then we give the result of both the greedy and the tabu search (TS-1) algorithms according to this upper bound. The running time comparison of both algorithms are also given.

The results of reconfiguration algorithms depend not only on the algorithms themselves but also on the input matrix. The best-fit strategy strives to place as much traffic as it can, but there is no guarantee to fit all of the traffic in. Here we first develop an upper bound U on the number of connections that could possibly be groomed:

$U = \sum_i \sum_j \sum_c \{V_{ij} \mid$ there is enough capacity between i and j for new traffic $T'[i][j]$ and there is an SADM at both nodes i and j on circle $c\}$.

The upper bound is computed by searching all the circles that are already built according to the old matrix to find available capacity to groom new connections. A connection can be established if there is capacity available between the terminal nodes and there are SADMs at the two nodes. The capacity of a connection is equal to the base bandwidth of one wavelength..

Although this upper bound is loose, because not all the connections available in the upper bound can be established at the same time, we can prove that no more connections can be built beyond this upper bound.

We define the load factor α by using this upper bound:

$$\alpha = \frac{\text{actual new connections established}}{\text{upper bound}} \times 100\%$$

This factor shows how much percentage of the new traffic could be groomed into the current configuration according to the upper bound.

Table I shows the results for the greedy algorithm and TS-1 under different numbers of nodes and different granularities for unidirectional rings ($r = 12$). For each entry, 20 matrix pairs (old matrix and new matrix) are randomly generated. The average value of α is computed. We observe that tabu search yields better results than the greedy algorithm for most cases, especially with a large number of nodes and large granularity. Table II shows the average number of new connections that can be groomed by the greedy algorithm and the tabu search algorithm using the same set of input data as Table I. The tabu search heuristic gains 2% more connections than the greedy algorithm on the average.

Here we give a specific example to indicate that the tabu search

Node number	method	g=3	g=4	g=12
N=5	greedy	92	89	79
(α *in* %)	tabu	92	90	80
N=6	greedy	90	89	78
(α *in* %)	tabu	91	90	80
N=8	greedy	90	84	76
(α *in* %)	tabu	91	85	77
N=12	greedy	86	83	70
(α *in* %)	tabu	87	84	72
N=16	greedy	85	82	67
(α *in* %)	tabu	86	84	68
N=20	greedy	84	80	63
(α *in* %)	tabu	86	82	69

TABLE I

RESULTS FOR THE BEST-FIT ALGORITHMS. N IS THE NUMBER OF NODES IN A RING. α IS THE LOAD FACTOR. g IS THE GRANULARITY.

Number of Nodes	method	g=3	g=4	g=12
N=5	greedy	16	17	22
	tabu	17	17	23
N=6	greedy	22	24	35
	tabu	22	25	36
N=8	greedy	39	40	54
	tabu	39	40	54
N=12	greedy	78	87	123
	tabu	79	88	126
N=16	greedy	128	142	193
	tabu	130	146	198
N=20	greedy	204	224	280
	tabu	209	229	290

TABLE II

ABSOLUTE CONNECTIONS BUILT BY THE GREEDY AND THE TABU ALGORITHMS.

heuristic can find a better solution than the greedy algorithm. In this example, there are 5 nodes in a unidirectional SONET ring. The granularity of the ring is 3. The old traffic matrix and the new matrix are generated as follows:

$$T = \begin{bmatrix} 0 & 2 & 6 & 1 & 0 \\ 7 & 0 & 6 & 1 & 3 \\ 2 & 2 & 0 & 3 & 3 \\ 5 & 2 & 7 & 0 & 2 \\ 4 & 5 & 7 & 2 & 0 \end{bmatrix} \quad T' = \begin{bmatrix} 0 & 4 & 6 & 2 & 5 \\ 7 & 0 & 8 & 1 & 4 \\ 3 & 3 & 0 & 10 & 11 \\ 5 & 4 & 7 & 0 & 2 \\ 6 & 5 & 7 & 3 & 0 \end{bmatrix}$$

Figures 1 and 2 show the grooming results for the greedy algorithm and the tabu search algorithm respectively. The following two matrices (R_g, R_t) are the remaining traffic matrices that cannot be groomed after employing those two algorithms. We observe from Figure 2 that two connections of $t_{2,3}$ are groomed to wavelength 5 instead of wavelength 4 in the tabu search algorithm. Then two more connections of $t_{2,4}$ could be groomed to wavelength 4 which results in a gain of 2 more connections for the tabu search than the greedy heuristic.

$$R_g = \begin{bmatrix} 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 2 \\ 0 & 2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix} \quad R_t = \begin{bmatrix} 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

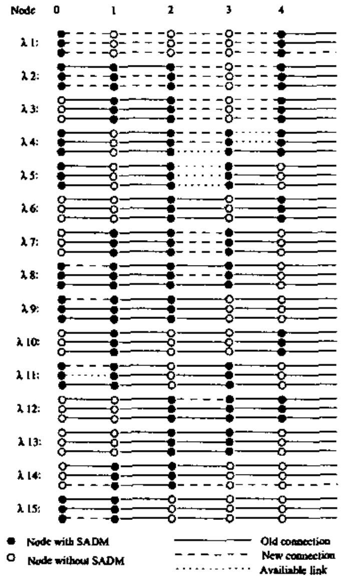

Fig. 1. An example of a reconfiguration problem using the greedy algorithm proposed in this work.

While tabu search yields better results in most cases, it also takes more time to obtain the solution with the tabu heuristic than with the greedy algorithm. Figure 3 shows the running time of the algorithms for the unidirectional ring under different values of granularity and different numbers of nodes. It uses the same set of data that generated the result in Table I. As we note from the figure, the running time increases greatly when the number of nodes in the ring increases. But if the cost of a single link fiber is very high, it is still worthwhile to employ tabu search algorithm compared to the greedy algorithm to earn more revenue.

B. Results for the full-fit case

We mentioned in Section III-B.2 that the static traffic grooming problem is a special case of the dynamic problem (for the full-fit case). We run the algorithm (TS-2) for the static traffic grooming problem for a unidirectional ring under uniform traffic. The traffic amount of each entry in the matrix is one unit (equals base bandwidth). We compare the results with both the greedy algorithm [6] and the simulated annealing algorithm [5] in Table III. We observe that the tabu search algorithm obtains the same results as simulated annealing (SA) for most cases. One entry for the tabu search method is better than SA. For some cases, the number of SADMs used in tabu heuristic is a little bit more than in SA. The running time for each entry is less than

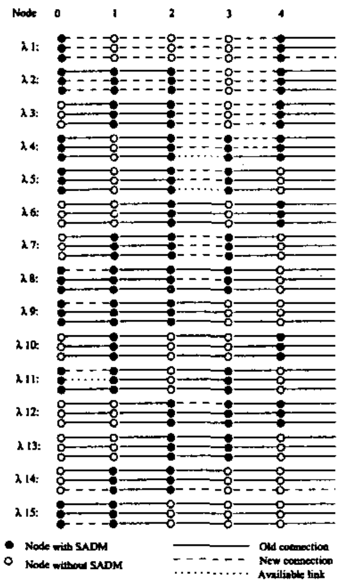

Fig. 2. An example of a reconfiguration problem using the tabu search heuristic (TS-1) proposed in this work.

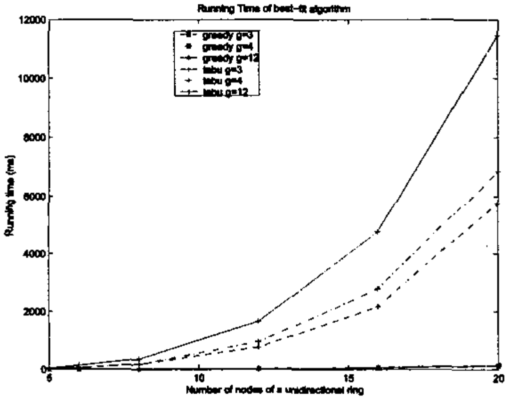

Fig. 3. Running time of the best-fit algorithm.

27 seconds on a 450 MHz UltraSPARC II processor based SUN Ultra-60 Workstation. In [5], the SA algorithm was run for 30 trials and best result was chosen. TS-2 does not depend on a statistical result and need not be run multiple times. We observe that it is never worse than the greedy algorithm. TS-2 is relatively stable, which is preferred in dynamic models. Because the traffic pattern changes from time to time, it is usually not feasible to run the program many times to obtain the best solution.

V. Conclusion

In this work, we introduced the dynamic traffic grooming model for reconfiguration problems. Two cases (best-fit and full-fit) are presented in an integer linear programming description. Since integer linear programming problems are NP-hard, we expect that dynamic traffic grooming problem is also intractable for both cases. For the best-fit case, two heuristic algorithms greedy and tabu search (TS-1) are proposed. An upper bound of the number of new virtual connections is developed to evaluate the performance of the algorithms. The results show that the tabu search algorithm (TS-1) proposed in this study will yield better solutions but takes more running time than the greedy algorithm for the best-fit case. For the full-fit case, a two-phase algorithm is developed. We observe that the static traffic grooming problem is a special case of the dynamic traffic grooming problem. The algorithms we proposed here (particularly the full-fit case, TS-2) can also solve the static grooming problem. Our algorithm is more stable than the simulated annealing algorithm proposed in previous work. Moreover, some of the solutions are better than those obtained in the earlier work on the static grooming problem.

g	method	N=4	N=8	N=16	N=20
g=3	greedy	7	31	79	146
	SA	7	31	69	124
	TS-2	7	31	69	124
g=4	greedy	7	29	71	123
	SA	7	28	66	120
	TS-2	7	28	66	120
g=16	greedy	4	14	36	65
	SA	4	14	33	57
	TS-2	4	14	33	58
g=48	greedy	4	8	21	34
	SA	4	8	19	37
	TS-2	4	8	21	32
g=64	greedy	4	8	15	28
	SA	4	8	15	28
	TS-2	4	8	15	28

TABLE III
Results for static traffic grooming.

References

[1] Angela L. Chiu and Eytan Modiano, "Reducing Electronic Multiplexing Costs in Unidirectional SONET/WDM Ring Networks via Efficient Traffic Grooming", *IEEE/IEICE Global Telecommunication Conference*, vol. 1, pp. 332-327, Sydney, Australia, November 1998.

[2] Ori Gerstel and Rajiv Ramaswami, "Cost Effective Traffic Grooming in WDM rings", *Proceedings of IEEE INFOCOM '98*, vol. 1, pp. 69-77, 1998.

[3] Eytan Modiano and Aradhana Narula-Tam, "Mechanisms for providing optical bypass in WDM-based networks", *SPIE Optical networks magazine*, January 2000.

[4] Alexander Schrijver, *Theory of Linear and Integer Programming*, John Wiley & Sons, New York, 1986.

[5] Jian Wang, V. Rao Vemuri, Wonhong Cho and Biswanath Mukherjee, "Improved Approaches for Cost-effective Traffic Grooming in WDM Ring networks: Non-uniform Traffic and Bidirectional Ring", *Proceedings ICC 2000*, 2000.

[6] Xijun Zhang and Chunming Qiao, "An Effective and Comprehensive Approach for Traffic Grooming and Wavelength Assignment in SONET/WDM Rings", *Proceedings of the SPIE-The International Society for Optical Engineering*, vol. 3531, pp. 221-232, 1998.

[7] Randall Berry and Eytan Modiano, "Reducing electronic multiplexing costs in SONET/WDM rings with dynamically changing traffic", *IEEE Journal on Selected Areas in Communications*, vol. 18, No. 10, October, 2000, pp. 1961-1971.

[8] Peng-jun Wan, Liwu Liu and Ophir Frieder, "Grooming of Arbitrary Traffic in SONET/WDM Rings", *IEEE/IEICE Global Telecommunication Conference*, vol. 1B, pp. 1012-1016, 1999.