Efficient Solution of the Traffic Grooming Problem in Light-Trail Optical Networks

Amr S. Ayad
Faculty of Computers & Information Tech., Cairo University
a.saad@fci-cu.edu.eg

Khaled M. El Sayed
Dept. of Electronics & Comm. Engineering, Cairo University
khaled@ieee.org

Sanaa H. Ahmed
Faculty of Computers & Information Tech., Cairo University
s.hanafy@fci-cu.edu.eg

Abstract

The concept of light-trails has been introduced in [1] as an efficient and feasible technology for IP transport over all-optical networks. Light-trails have been shown to offer a number of advantages over other technologies such as Optical Burst Switching (OBS) or Optical Packet Switching (OPS). This paper discusses the routing problem of light-trails. We present an enhancement for reducing the size of the routing problem before optimally solving it using Integer Linear Programming (ILP). We also propose an efficient heuristic for solving the static routing problem. The results show the efficiency of the proposed heuristic scheme.

1. Introduction

Present DWDM transport networks are circuit-based backbones used to transport TDM, ATM, Ethernet, and IP services as overlay networks. This architecture served well as a multi-service transport technology with mature standards in place.

The emergence of converged IP-based services such as triple-play and IP Multimedia Subsystem (IMS) [2] has urged carriers to exploit the benefits of a consolidated IP/MPLS backbone that would bring down both of their capital and operational expenses.

A need has, therefore, emerged for a transport technology that would achieve an efficient use of all-optical networks when carrying IP traffic.

A number of frameworks have been proposed targeting a packet-based transport technology that would still make efficient use of the inherent circuit-switching nature of wavelength channels offered by DWDM. These are mainly: wavelength routing networks (lightpaths) [3], Optical Burst Switching (OBS) [4], and Optical Packet Switching (OPS) [5].

In [6], light-trails framework was shown to be superior to lightpaths, OBS, and OPS in terms of being technologically feasible, allowing faster service provisioning, grooming of sub-wavelength demands, and adapting to the bursty nature of IP traffic.

Light-trails are based on the use of a drop-and-continue sharing scheme of a wavelength channel. A light-trail is formed by signaling a number of consecutive nodes on a network path to share a certain wavelength. Upstream nodes can transmit to downstream nodes but not vice versa. The description below is per wavelength and the mentioned components are to be replicated for each wavelength carried on a physical network link.

Using a drop coupler, each node couples a portion of the source's optical energy into its own local receiver. Using an add coupler, any node on the light-trail can transmit traffic to its downstream neighbors.

Each node is equipped with an optical shutter that can be either switched on or off to either allow the optical signal to further propagate downstream or to be blocked at the given node. The nodes at the two ends of a light-trail configure their shutters to the off state for the whole lifetime of a light-trail such that the signals transmitted by light-trail nodes remain confined to it and thus allowing for spatial re-use of the wavelength channel by other light-trails.

Figures 1 and 2 from [1] illustrate the concept.

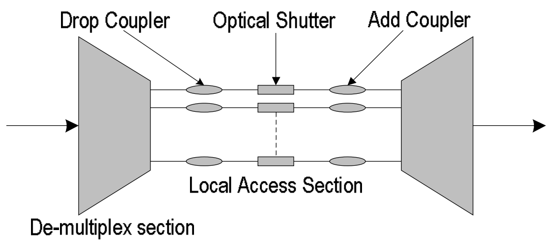

Figure 1: Architecture of a light-trail node

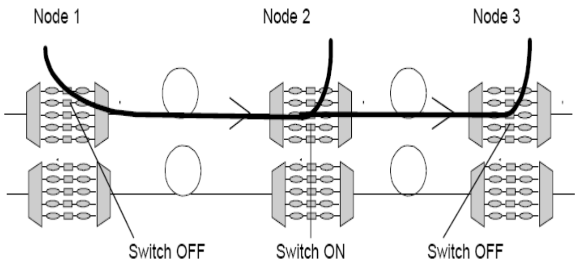

Figure 2: Three nodes in a light-trail configuration

Establishing light-trails has a set of constraints. The total bandwidth demand of carried flows on a certain light-trail must not exceed the capacity of the wavelength channel. Another constraint is the hop count; the strength of a light-trail signal tends to degrade more rapidly than lightpaths as the part of its energy is dropped at each node. It has been shown in [1] that 6 hops would still result in an acceptable received signal in a metro environment. In [6], a pre-processing algorithm that re-arranges the given traffic matrix to handle flows whose shortest path from source to destination still has a higher hop count. This pre-processing will result in a modified traffic matrix that is input to the routing problem which remains, basically, the same. Therefore we only consider the routing problem with a given traffic matrix whether the pre-processing was done or not.

The dynamic routing problem is one in which no knowledge of future demands is assumed when making the routing decision for the newly requested flow. This paper tackles the problem of static routing of light-trails. The static routing problem is one in which all flows are known before routing is performed.

The rest of the paper is organized as follows: in section 2, an example is given to present the problem terminology to be used throughout the paper. In section 3, we give the approach used to solving the light-trail routing problem. In section 4, we discuss the ILP solution of the static problem as formulated in [6] and propose an enhancement to reduce the problem size and hence its solution time. In section 5, we discuss the proposed heuristic as applied to the static routing problem. The results of applying our heuristic to example problems are given in section 6. Section 7 discusses future work and concludes the paper.

2. Problem terminology

We use the topology and traffic matrix in Figure 3 and Table 1 respectively to describe the used terminology for the rest of the paper as well as the approach used to solve the routing problem.

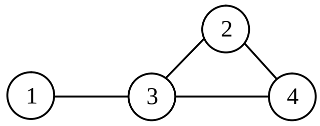

Figure 3: An example network topology

Table 1: An example demand matrix

From/To	1	2	3	4
1	0	5	2	7
2	10	0	17	8
3	30	20	0	35
4	0	5	11	0

Solving the routing problem aims to assign each offered flow to a light-trail. For the rest of this paper; we assume the unit of the traffic demand to be equivalent to an OC-1 signal (51.84 Mbps) and we also assume all wavelength channels to be equivalent to an OC-48 signal (2.48 Gbps). Therefore, the capacity of a wavelength channel is equal to 48 traffic demand units.

We define the set of *eligible paths for a flow* as the paths that include both the end nodes of a flow with the source node upstream with respect to the destination node. For example, the eligible paths for the flow (2,1) are (2,3,1) and (2,4,3,1).

We define the set of *eligible flows for a path* as the flows with both end nodes belonging to the path with the source node upstream with respect to the destination node. For example: the eligible flows for the path (2,4,3,1) are (2,4), (2,3), (2,1), (4,3), and (3,1).

We define a path to be *saturable* if the sum of traffic demands of its eligible flows exceeds its capacity. For example, the eligible flows of path (3,2,4) are (3,2), (3,4), and (2,4) with total traffic demand of 63 (> 48). Therefore, the path (3,2,4) is said to be saturable; while flow (1,3) is the only eligible flow for path (1,3) with a traffic demand of 2 (<48). Therefore, the path (1,3) is not saturable.

Different light-trails sharing the same network link must use different wavelengths. Each light-trail occupies the same wavelength on all links between its start and end nodes.

Finally, we note that maximum grooming is achieved if every occupied wavelength channel is used to its full capacity. A simple lower bound to the number of light-trails that need to be established to carry all of the offered flows is then:

$$MinNumLTs = \sum_{f \in F} D_f \Big/ C \qquad (1)$$

Where C is the capacity of a wavelength channel in traffic demand units, and F is the set of offered flows.

3. Solution approach

Figure 4 outlines the overall approach for solving the light-trail routing problem.

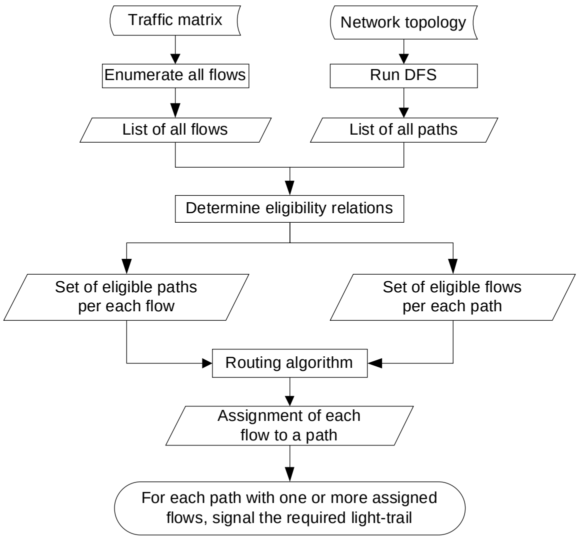

Figure 4: Overall solution approach

As proposed in [6], we use DFS to enumerate all possible network paths up to a certain hop limit. The eligibility relations among enumerated paths and flows are then determined to generate a set of eligible paths for each flow and a set of eligible flows for each path.

The used routing algorithm can then be based on either solving an ILP to obtain optimal results or using some heuristic that provides near-optimal results but at much less computational cost. Our contribution includes an enhancement to the ILP formulation proposed in [6] and a routing heuristic that reaches a near-optimal solution.

The objective of the routing algorithm is to enhance a certain performance metric such as the total number of assigned light-trails or the total number of wavelength links in use.

4. ILP solution

The ILP formulation of the light-trail routing problem in [6] aimed at minimizing the number of light-trails signaled to carry the given demand matrix. This minimization implies better grooming of the fractional demands into light-trails and therefore a more effective utilization of the occupied wavelengths.

The problem formulation in [6] is given as follows:

- Parameters:

For the given directed graph $G(V,E), N=|V|$, let LT be the set of all the possible light trails within hop-length limit $L_{\max}$, and $l = 1, 2, \dots, |LT|$ be the number assigned to each light trail in the LT. Let C denote the full-wavelength capacity which is represented as an integer which is a multiple of smallest capacity requests. The smallest capacity request is denoted as 1. The integer entry in traffic matrix $T_{N\times N}$ represented by t_{ij}, denotes the requested capacity from node i to node j, $t_{ij} \le C$ as we are only considering demands of fractional wavelength capacity to assess grooming efficiency of light trail networks.

- Variables:

m_{st}^{l} : binary variable, route indicator, takes value of 1 if request (s, t) takes light trail l; zero otherwise.

d^{l} : binary variable, light trail usage indicator, takes value of 1 if trail l is used by any request; zero otherwise.

- ILP Formulation:

Objective: $$min \sum_{l} c^{l} \times d^{l} \qquad (2)$$

When $c^l = 1$, the objective is to minimize the number of light-trails in use.

Assignment Constraints: Each request is assigned to one and only one light trail.

$$\sum_{l} m_{st}^{l} = 1 \quad \forall (s,t): t_{st} \in T,\ t_{st} > 0 \qquad (3)$$

Light-trail capacity constraints: The aggregate request capacity on a light trail should not exceed the full wavelength capacity.

$$\sum_{(s,t)} m_{st}^{l} t_{st} \le C \qquad (4)$$

Light-trail usage constraints: If any flow is assigned on light trail l, δ^l is set to 1; otherwise, if none of the flows picked light-trail l, $\delta^l = 0$. Recall that δ^l is a binary variable.

$$d^{l} \ge m_{st}^{l} \quad \forall \quad (s,t): t_{st} \in T \qquad (5)$$

$$d^{l} \in \{0,1\} \qquad (6)$$

4.1. Proposed modification

We propose an enhancement to the ILP formulation in [6] by making use of the saturable path definition given in section 2. If a path is not saturable, then there is no need to include a capacity constraint for it in the ILP formulation. This can greatly reduce the number of capacity constraints. Using the modified ILP formulation, we were able to reduce the solution time of the routing problem in [6] from 2146 seconds to 1117 seconds (about 48% less). The solution was obtained using the open source GLPK library [7] on a Pentium M 2 GHz processor with 1 GB of RAM.

It was noted in [6] that the ILP solution tends to prefer paths with more hops to be assigned as light-trails because this would allow for more flows to share the same light-trail. This trend, however, comes at the expense of consuming more wavelength links. A proposed extension to the work in [6] was to use an objective function that would minimize the number of wavelength links in use rather than the number of light-trails. This can be done by re-defining the value of the parameter c^l in (1) to become the hop-length of the light-trail. We have performed such modification of the objective function and actually obtained a routing solution that reached the same number of light-trails as the original ILP but with a lower number of occupied wavelength links (49 wavelength links versus 52 wavelength links for the original problem).

5. Proposed heuristic

The objective of the proposed heuristic, shown in Figure 5, is to route offered demands using the least number of light trails. The complexity of the heuristic is $O(N^2 \log N)$. We omit the details of the complexity analysis due to space limitation. The algorithm uses a number of attributes, shown in Table 2.

Table 2: Attributes used with the proposed heuristic

Path Attributes	
Static Attributes	Dynamic Attributes
EF_p : Number of eligible flows for path ***p***.	RF_p: Number of flows routed on path ***p***.
ED_p: Eligible demand for path ***p*** which is the sum of traffic demands of the eligible flows for path ***p***.	RD_p: Routed demand for path ***p*** which the sum of traffic demands of the flows routed on path ***p***.
Flow Attributes	
EP_f :Number of eligible paths for flow *f*.	
D_f :Demand of flow *f*.	

We define ***static*** attributes as the attributes whose values are known prior to running the routing algorithm. Their values do not change as flows are being routed. *Dynamic* attributes are continually updated during the course of the algorithm. Our algorithm uses the above attributes to route each flow (one-by-one) on one of its eligible paths.

The proposed heuristic involves two sorting operations of the list of flows and the list of eligible paths. These are needed to make two decisions per iteration until all flows are routed. The first decision is to select the next flow to be routed using the flow attributes. The second decision is to select the path on which the flow is routed using the path attributes. Sections 5.1 and 5.2 outline the two sorting operations.

5.1. Sorting the list of flows

Given the list of flows to be routed, our heuristic computes the values of flow attributes for each flow. It then sorts the list of flows in a way that results in maximum sharing of light-trails among routed flows.

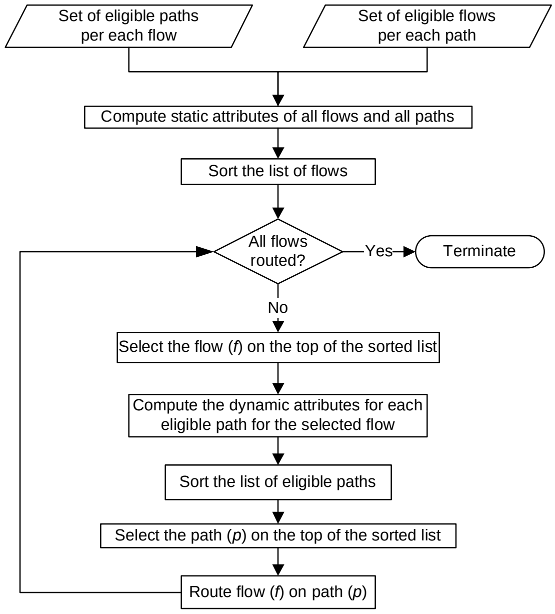

Figure 5: Flowchart of the proposed heuristic

Considering the demand attribute (D_f), if flows with lower values of demand were routed first; then it is more likely for flows with higher demand to be routed later on separate light-trails rather than sharing the already occupied light-trails. Already occupied light-trails are less likely to have enough capacity to accommodate such flows. Less sharing of light-trails causes the final routing solution to use a larger number of light-trails which is not desirable. We thus choose to sort the list of flows in the descending order of D_f and select the flow on top of the list.

Considering the number of eligible paths attribute (EP_f), this attribute represents the allowed degree of freedom when making the routing decision of a flow. If $EP_f = 1$ for a flow (*f*), then it can only be routed on a certain path irrespective of whether this path is shareable with other flows or not. It is thus preferable to start routing such flows with a limited degree of freedom. This would later allow flows with more eligible paths to prefer the already occupied paths for better sharing of light-trails. We thus choose to sort the list of flows in the ascending order of EP_f and select the flow on top of the list.

To reduce the computational complexity of the sorting function, we combine both flow attributes using a weighted sum in a single *flow-preference* Q_f attribute:

$$Q_f = W_D \times D_f - W_{EP} \times EP_f \qquad (7)$$

Where $W_D, W_{EP} > 0$

A higher value of Q_f implies a higher value of the D_f and a lower value of the EP_f. Therefore, sorting the flow list in the descending order of Q_f achieves the above sorting objectives of both flow attributes.

Setting the values of weights W_D and W_{EP} allows the algorithm to determine which attribute (D_f or EP_f) to be used as the primary sorting attribute which is critical to the quality of the obtained routing solution. We propose a rule for making this decision by computing:

$$EP_{\min} = \min_{f \in F} EP_f \quad (8)$$

We choose D_f to be the primary sorting key of the flow list if $EP_{\min}$ is higher than *MinNumLTs*. Because if $EP_{\min}$ is higher than the minimum number of routes used to carry the offered flows; then most flows have a sufficient number of alternate paths to be routed on; and the D_f attribute should be the primary sorting key.

To use any attribute as the primary sorting key, its weight must be higher than the maximum difference between any two values of other attributes.

If D_f is to be used as the primary sorting key (that is $EP_{\min} > MinNumLTs$), the weights are calculated as:

$$W_{EP} = 1,\ W_D = \Delta\, EP_{\max} + 1 \quad (9)$$

Where:

$$\Delta\, EP_{\max} = EP_{\max} - EP_{\min}\ ,\ \ EP_{\max} = \max_{f \in F} EP_f$$

If EP_f is to be used as the primary sorting key (that is $EP_{\min} < MinNumLTs$), the weights are calculated as:

$$W_D = 1,\ W_{EP} = \Delta\, D_{\max} + 1 \quad (10)$$

Where:

$$\Delta\, D_{\max} = D_{\max} - D_{\min}\ ,$$

$$D_{\max} = \max_{f \in F} D_f\ ,\ \ D_{\min} = \min_{f \in F} D_f$$

5.2. Sorting the list of eligible paths

Having selected the flow to be routed next; it then needed to select an eligible path with enough spare capacity. The list of eligible paths must be sorted using the path attributes listed in table 2.

Considering a certain path i, a higher value of ED_i and EF_i indicates that the path is more likely to be selected for future flows implying a better opportunity of capacity sharing. A higher value of RD_i and RF_i indicates that the path has been already used to route more past flows; again implying more preference for selecting it to boost capacity sharing.

We combine the four path attributes using a weighted sum in one path-preference Q_p attribute:

$$Q_p = W_{RD} \times RD_p + W_{RF} \times RF_p + W_{ED} \times ED_p + W_{EF} \times EF_p \quad (11)$$

where: W_{RD}, W_{RF}, W_{ED}, $W_{EF} > 0$

A higher value of Q_p implies a higher value for all path attributes. Therefore, sorting the eligible path list in the descending order of Q_p achieves the preference objectives of path selection.

Our results have shown no significant effect of the attributes based on the number of flows (EF_p and RF_p). Using a value of unity for W_{EF} and W_{RF} had no effect on the obtained results. This is consistent with the fact that network capacity is limited in terms of the channel bandwidth rather than the number of flows per channel.

On routing the initial few flows, the RD_p attribute has a zero value for most paths; the value of ED_p will then solely determine the routing decision. As the solution progresses with more flows being routed, the RD_p attribute begins to affect the routing decision to a degree that is relative to the value of W_{RD}.

Intuitively, the algorithm should always prefer paths with more used capacity when routing a new flow. However, when routing the first few flows, the used capacity is still zero for most paths. Using random tie breaks for initial routing decisions degrades the quality of the final results as the algorithm tends to pack all future flows on randomly selected routes. It is thus critical to exploit the amount of eligible traffic attribute for the initial routing decisions. Our algorithm adopts to the former consideration by setting the weights (W_{ED} and W_{RD}) to allow the ED_p attribute to be the primary sorting key for initial flows and then switches to using the RD_p attribute as the primary sorting key once it begins to have non-zero values. Thus, the weights are: (10)

$$W_{RF} = W_{EF} = W_{ED} = 1,\ W_{RD} = \Delta\, ED_{\max} + 1 \quad (12)$$

Where:

$$\Delta\, ED_{\max} = ED_{\max} - ED_{\min},$$

$$ED_{\max} = \max_{p \in LT} ED_p\ ,\ \ ED_{\min} = \min_{p \in LT} ED_p\ ,$$

6. Results

Here we present the results of using the proposed heuristic. We have tested two example problems that were previously solved in [6] and [8].

6.1. First problem: 6-node mesh network

Below are the network topology and traffic matrix used in [8]. We assume a hop limit of three hops. The optimal ILP solution routes the total demand on 10 light-trails. Using the proposed heuristic, we were able to route the same demand on 11 light-trails. Table 4 provides the routing result.

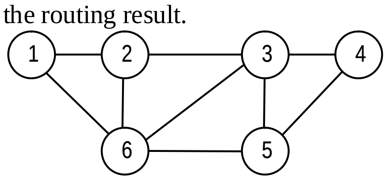

Figure 6: A 6-node mesh network from [8]

Table 3: Traffic matrix for 6-node network

From/To	1	2	3	4	5	6
1	0	7	6	9	19	17
2	27	0	3	6	28	2
3	14	3	0	19	31	9
4	26	5	29	0	5	23
5	27	20	20	17	0	14
6	9	30	1	1	1	0

Table 4: Routing result of the 6-node network of figure 6

Path	Carried Flows	Load
(3,6,2,1)	(3,6)(3,2)(6,1)(2,1)	48
(4,3,5,6)	(5,6)(4,3)(4,5)	48
(1,6,3,2)	(6,3)(6,2)(1,6)	48
(1,2,3,5)	(2,3)(3,5)(1,2)(1,3)	47
(1,6,5,4)	(6,5)(5,4)(6,4)(1,5)(1,4)	47
(5,6,2,1)	(5,2)(5,1)	47
(4,3,6,1)	(3,1)(4,1)	40
(2,3,5,4)	(2,5)(2,4)	34
(4,3,6,2)	(4,6)(4,2)	28
(5,3,2,6)	(2,6)(5,3)	22
(3,4)	(3,4)	19

6.2. Second problem: 10-node mesh network

Below are the network topology and traffic matrix used in [6]. We assume a hop limit of four hops. The optimal ILP solution routes the total demand over 13 light-trails. Using the proposed heuristic, the offered traffic was routed over 20 light-trails. However, more than 94% of the traffic amount was routed over the same optimal number of 13 light-trails. Table 6 provides the routing results. We note, by manual inspection of the results, that the routing decisions for some flows did not well serve the purpose of the algorithm. For example, flows (3,1) and (4,2) could have been routed on a single light trail (4,3,2,1). This is an expected “glitch” of the algorithm due to the random component associated with the initial routing decisions as discussed on section 5.1.

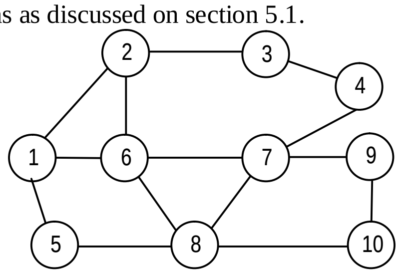

Figure 7: A 10-node mesh network from [6]

Table 5: Traffic matrix for 10-node network

From/To	1	2	3	4	5	6	7	8	9	10
1	0	5	11	10	4	5	4	6	6	10
2	8	0	5	5	1	3	1	11	7	2
3	3	0	0	3	0	2	9	4	10	9
4	8	2	11	0	11	6	11	6	9	4
5	11	7	7	6	0	11	3	2	9	9
6	8	9	7	5	4	0	11	8	10	9
7	9	7	11	9	1	10	0	4	11	2
8	6	0	10	4	2	4	4	0	2	9
9	2	9	10	2	6	9	9	8	0	9
10	11	0	10	0	8	10	8	11	4	0

Table 6: Routing result of the 10-node network in figure 7

Path	Carried Flows	Load
(5,1,6,7,9)	(1,6)(1,7)(5,6)(5,7)(6,9)(1,9)(5,9)	48
(5,1,6,8,10)	(5,1)(8,10)(6,10)(1,10)(5,10)	48
(9,7,6,2,3)	(7,2)(2,3)(6,3)(9,6)(9,2)(9,3)	47
(9,10,8,5,1)	(8,5)(9,8)(9,10)(10,1)(9,1)(10,5)(9,5)	46
(10,8,7,4,3)	(4,3)(7,3)(8,4)(8,3)(10,3)	46
(9,10,8,6,7)	(8,6)(8,7)(10,8)(9,7)(10,7)(10,6)	46
(3,4,7,9,10)	(7,10)(3,4)(3,7)(4,10)(4,9)(3,9)(3,10)	46
(2,6,8,7,9)	(6,7)(2,6)(7,9)(2,8)(2,7)(8,9)(2,9)	46
(3,4,7,6,8)	(6,8)(7,6)(7,8)(4,7)(3,6)(4,8)(3,8)	45
(4,7,6,1,5)	(1,5)(7,1)(6,5)(7,5)(4,6)(4,1)(4,5)	43
(5,1,6,2,3)	(6,2)(1,2)(5,2)(1,3)(5,3)	39
(5,1,6,7,4)	(7,4)(6,4)(1,4)(5,4)	30
(2,1,6,8,10)	(1,8)(2,1)(2,10)	16
(2,6,8,5,1)	(6,1)(8,1)(2,5)	15
(2,3,4)	(2,4)	5
(10,9)	(10,9)	4
(3,4,7,6,1)	(3,1)	3
(4,7,6,2)	(4,2)	2
(9,10,8,7,4)	(9,4)	2
(5,8)	(5,8)	2

7. Conclusion and future work

We have proposed an enhancement to reducing the number of constraints of the ILP formulation of light-trail routing problem. We have also proposed a heuristic that is based on multi-attribute sorting of both the set of offered flows and the set of available network paths. Our heuristic gives near-optimal results in terms of the number of used light-trails to carry the offered traffic.

Our future work includes porting our algorithm to the survivable case where each flow is assigned to two link-disjoint paths for protection against link failures.

8. References

[1] A. Gumaste, “Light-Trails and Light-Frame Architectures for Optical Networks,” Ph.D. thesis, Fall 2003, University of Texas at Dallas, www.cotrion.com/light-trails

[2] B. Zhu, “Analysis Of SIP in UMTS IP Multimedia Subsystem”, M. Sc. Thesis, Computer Engineering, NC State University, 2003.

[3] Z. Zhang, J. Fu, D. Guo, L. Zhang, “Lightpath Routing for Intelligent Optical Networks,” *IEEE Network*, pp. 28-35, July/August 2001.

[4] C. Qiao and M. Yoo, “Optical Burst Switching (OBS) – A New Paradigm for an Optical Internet,” *J. High Speed Networks*, pp. 69–84, vol. 8, 1999.

[5] M. Mahony, D. Simeonidou, D. Hunter, and A. Tzanakaki, “The application of optical packet switching in future communication networks,” *IEEE Communication Magazine*, pp. 128–135, March 2001.

[6] J. Fang, W. He, and A. K. Somani, "Optimal Light Trail Design in WDM Optical Networks," In *Proceedings of IEEE ICC’04*, pp. 1699-1703, volume 3, June 2004. www.public.iastate.edu/~jinger23/Research/Jing-2004-ICC.pdf

[7] Andrew Makhorin, GNU Linear Programming Kit (GLPK), Dept. for Applied Informatics, Moscow Aviation Institute, http://www.gnu.org/software/glpk/glpk.html

[8] J. Fang, W. He and A. K. Somani, “IP Traffic Grooming in Light Trail Optical Networks,” www.public.iastate.edu/~jinger23/resource/Publications.htm.