Enhanced Optimal and Heuristic Solutions of the Routing Problem in Light Trail Networks

Amr S. Ayad, Khaled M. ElSayed†, and Sanaa H. Ahmed

Abstract—**Light-trail is an efficient and feasible technology for IP transport over all-optical networks. The proposition of light trails for all-optical networks has demonstrated a number of advantages over other paradigms such as Wavelength Routing (WR), Optical Burst Switching (OBS), and Optical Packet Switching (OPS). This paper tackles the routing problem of light-trails with the solution objective of minimizing the number of needed light-trails to accommodate an offered traffic matrix. We present two enhancements to the Integer Linear Programming (ILP) formulation of the routing problem. We also propose a computationally efficient routing heuristic for use with static and incremental traffic models. The efficiency of the proposed heuristic is confirmed using example problems of different network topologies.**

Index Terms—**all-optical networks, IP over optical, light trails, traffic grooming**

I. 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], the framework of light-trails was shown to be superior to lightpaths, OBS, and OPS in terms of being technologically feasible, allowing for faster service provisioning time, grooming of sub-wavelength demands, and adapting to the bursty nature of IP traffic. A brief explanation of the technology is given below. More details are available in [1], [6], and [14].

Light-trails are based on the use of a drop-and-continue sharing scheme of a wavelength channel. Upstream nodes can transmit to downstream nodes but not vice versa. The following description per-wavelength; the mentioned components are replicated for each wavelength carried on a physical network link.

Using a drop coupler, a node couples a portion of the optical energy into its own local receiver. Using an add coupler, any node on the light-trail can transmit traffic to downstream neighbors; only the interested downstream node will process the incoming traffic.

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 (termed the convener and end nodes) 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 subsequently allowing for spatial re-use of the wavelength by other light-trails. Figures 1 and 2 illustrate the concept [1].

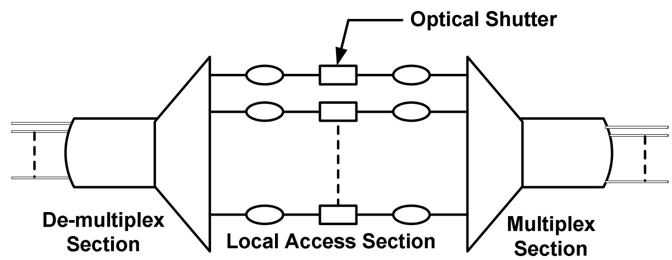

Fig. 1: Architecture of a light-trail node

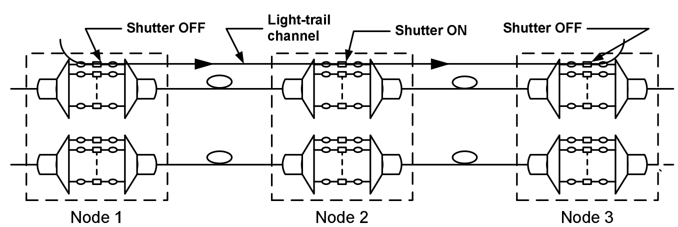

Fig. 2: Three nodes in a light-trail configuration

While the node architecture in Figures 1 and 2 is simple enough to illustrate the concept; it is only applicable for nodes with a connectivity degree of two and is thus only usable in line or ring topologies. Two other architectures of a light-trail node for use in mesh networks with higher connectivity degree are given in [10] using wavelength-selective power blockers, and in [11] using optical space switches.

Bandwidth arbitration on the light-trail can be either stochastic [12] or static TDM (round robin or weighted round robin) [13]. In this paper, we assume static TDM arbitration with the transmission time allocated to each node on the light-trail proportional to its demand. On routing several flows on a certain light-trail, we put a constraint that the sum of their traffic demands does not exceed the capacity of the light-trail wavelength channel.

† Corresponding author (khaled@ieee.org)
Khaled M ElSayed is with the Faculty of Engineering, Cairo University. Amr S. Ayad and Sanaa H. Ahmed are with the faculty of Computers & Information Technology Cairo University.

The paper tackles the problem of routing flows on a light-trail network in a way that maximizes the grooming of sub-wavelength demands by reducing the number of signaled light-trails.

Better grooming in light-trail networks not only implies more efficient use of network resources such as wavelengths and transceivers but also provides for considerably faster service provisioning time comparing to lightpath or OBS schemes. Routing a new flow on an existing light-trail does not involve any optical switching but merely reqires the exchange of out-of-band (OOB) control packets over the Optical Supervisory Channel (OSC) among the participant nodes of the already established light-trail to accommodate the traffic of the new flow.

In [11], electronic grooming of flows is allowed; that is a flow can traverse more than one light-trail with electronic grooming performed at a downstream node that is common to both light-trails. We do not consider the electronic grooming as it implies additional cost and precludes rate and protocol transparency. We only consider the grooming in the sense of routing flows in a way that packs as many flows as possible on a light-trail for better utilization of its wavelength.

Routing with a static traffic model is a problem in which all offered flows are given before a solution technique starts routing them. With an incremental traffic model, the routing problem is one in which no knowledge of future demands is assumed when making the routing decision for the newly requested flow. We assume for the incremental problem that the lifetime of accepted flows is very long such that no flows terminate before routing the whole set of given flows.

In the rest of the paper, we will refer to routing problem with static traffic model as the static problem. We will refer to the routing problem with incremental traffic model as the incremental problem.

The rest of this paper is organized as follows: in section II, we use a simple example to present the problem terminology to be used throughout the paper. In section III, we outline the general approach used to solving the light-trail routing problem. In section IV, we give the ILP solution of the static problem as formulated in [6]; we also present two enhancements to the problem formulation. In section V, we discuss the proposed heuristic as applied to the static and incremental problems. In section VI, we give the results of applying our heuristics to sample problems and compare their quality with ILP solution. Section VII discusses future work and concludes the paper.

II. Problem Terminology

We use the topology in Fig. 3 to describe the used terminology throughout the paper. A sample demand matrix is given by Table I.

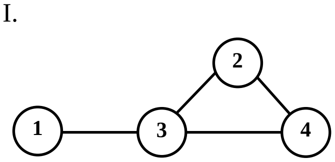

Fig. 3: An example network topology

TABLE I: 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

The capacity of a light-trail is equal to the speed of its wavelength channel. We assume the unit of traffic demand to be equivalent to the speed of an OC-1 signal (51.84 Mbps) and we also assume all wavelength channels to be having the same speed of OC-48 signal (2.48 Gbps). Therefore, the capacity of each wavelength channel is equal to 48 traffic demand units.

We define the set of *eligible paths for a flow* as the set of paths that includes both the source and destination nodes of a flow and in which the source node is upstream with respect to the destination node. The eligible paths for the flow (2→1) are (2 →3→1) and (2→4→3→1).

Similarly, we define the set of *eligible flows for a path* as the set of flows with both of their source and destination nodes belonging to the path with the source node upstream with respect to the destination node. 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. The eligible flows of path (3→2→4) are (3→2), (3→4), and (2→4) with a total traffic demand of 63 units (> 48). Therefore, the path (3→2→4) is said to be saturable; while the flow (1→3) is the only eligible flow for the path (1→3) with a traffic demand of 2 (<48). Therefore, the path (1→3) is not saturable.

Different light-trails passing by the same network link must use different wavelengths. Each light-trail occupies the same single wavelength on all links between its start and end nodes. Therefore, the number of occupied wavelength links by a light-trail is equal to its hop count. The total number of wavelength links occupied by a certain routing solution is equal to the sum of hop counts of the signaled light-trails.

Finally, we note that maximum grooming is achieved if every occupied wavelength channel is used to its full capacity. A simple, yet potentially loose, lower bound (*MinNumLTs*) to the number of light-trails that need to be established to carry the offered demand matrix can then be computed as:

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

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

III. Solution Approach

Fig. 4 outlines the overall approach that we follow for solving the light-trail routing problem.

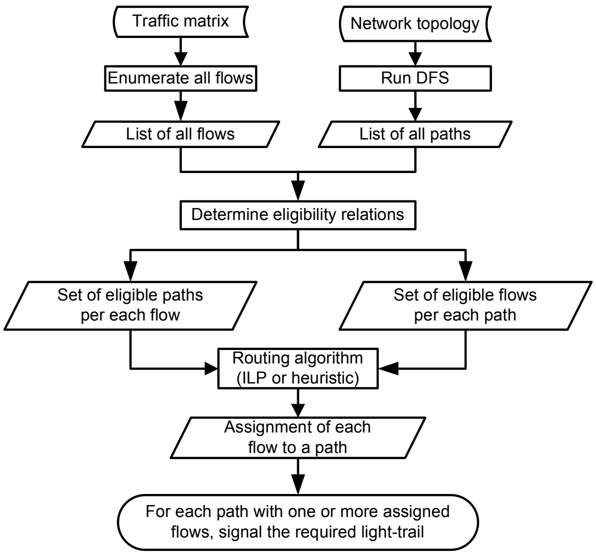

Fig. 4: Overall solution approach

As proposed in [6], we use Depth First Search (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.

For the static problem, the used routing algorithm can be based on either solving an ILP [1], [6], and [15] to obtain optimal results or using some heuristic that provides near-optimal results but at a much less computational cost. Our contribution includes two enhancements to the ILP formulation proposed in [6] and a routing heuristic that gives a near-optimal solution.

In contrast, it is computationally expensive to use ILP techniques for the incremental problem as an ILP needs then to be solved for each incoming flow. Therefore, we only use the proposed heuristic when solving the incremental problem.

The objective of the routing algorithm is to enhance a performance metric such as the total number of signaled light-trails or the total number of used wavelength links.

IV. ILP SOLUTION

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

- Parameters:

For the given directed graph $G(V,E), N = |V|$, let P be the set of all paths discovered by DFS within a hop limit of H, P_f to be the set of eligible paths for flow f, and F_p to be the set of eligible flows for path p. We assume $D_f \leq C$ for all flows as we are only considering demands of fractional wavelength capacity to assess the grooming efficiency of light-trail networks.

- Variables:

μ_f^p : binary variable, route indicator, takes a value of 1 if flow f takes path p; zero otherwise.

δ^p : binary variable, path usage indicator, takes a value of 1 if path p is used by any flow; zero otherwise.

- ILP Formulation:

Objective: $$min \sum_{p \in P} c^p \times \delta^p \tag{2}$$

When c^p = 1, the objective is to minimize the number of signaled light-trails to carry the offered traffic.

Assignment Constraints: Each flow is assigned to one and only one path.

$$\sum_{p \in P_f} \mu_f^p = 1 \quad \forall\ f \in F,\ D_f > 0 \tag{3}$$

Capacity constraints: The total amount of traffic routed on a light-trail should not exceed its wavelength capacity.

$$\sum_{F_p} \mu_f^p D_f \leq C \tag{4}$$

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

$$\delta^p \geq \mu_f^p \quad \forall\ \ f \in F, p \in P \tag{5}$$

$$\delta^p \in \{0,1\},\ \ p \in P \tag{6}$$

In sections IV-A, IV-B; we propose two enhancements to the above problem formulation.

A. Removing Redundant Constraints

We propose an enhancement to the ILP formulation in [6] by making use of the saturable path definition given in section II. 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 and, hence, the corresponding time for solving the ILP. Using the modified ILP formulation, we were able to reduce the solution time of the routing problem in [6] from 2,146 seconds to 1,117 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.

B. Narrowing the Search Space

We have noticed that the optimal solution of the initial relaxed LP gives a value of the objective function that is far below the expected optimal value of the ILP solution. Consequently, the subsequent B&B algorithm had more sub-problems to check in order to find the optimal solution and, potentially, to verify optimality even after reaching the objective value corresponding to the optimal integer solution.

Considering the lower bound in (1), we reduce the solution space by adding the below constraint to the problem formulation

$$\sum_{p \in P} c^p \times \delta^p \geq MinNumLTs \tag{7}$$

Upon adding the constraint (7); a large number of sub-problems were readily fathomed and it took relatively less running time for GLPK reach the integer optimal solution.

V. PROPOSED HEURISTIC

The objective of the proposed heuristic is to route the offered traffic using the least number of light-trails. The proposed heuristic operates on two steps. On the first step, the list of offered flows is sorted based on the below flow attributes. On the second step, the offered flows are routed one-by-one by the order determined in step 1. On routing a flow, the set of eligible paths with enough spare capacity is determined and then sorted based on the below path attributes to determine the light-trail over which the flow is routed. Fig. 5 shows the flowchart of the proposed heuristic.

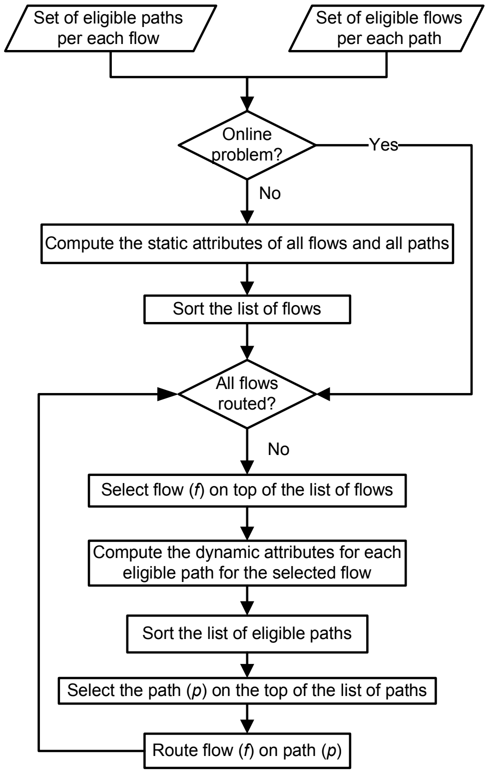

Fig. 5: Flowchart of the proposed heuristic

A. *Problem Attributes*

The algorithm makes use of a number of *static* and *dynamic* attributes. We define *static* attributes as the attributes whose values are known prior to running the routing algorithm. These values do not change as flows are being routed. Dynamic attributes are continually updated during the course of the algorithm upon the routing of each flow.
Following is a list of used attributes:

Static path attributes:

EF_p : Number of eligible flows for path *p*.

$$EF_p = \left|F_p\right| \tag{8}$$

ED_p: Eligible demand for path *p* which is the sum of traffic demands of the eligible flows for path *p*.

$$ED_p = \sum_{f \in F_p} D_f \tag{9}$$

Static flow attributes:

EP_f : Number of eligible paths for flow *f*.

$$EP_f = \left|P_f\right| \tag{10}$$

D_f : Demand of flow *f*.

Dynamic path attributes:

RF_p: Number of flows routed on path *p*.

RD_p: Routed demand for path *p* which the sum of traffic demands of the flows routed on path *p*.

For the static problem; the algorithm makes use of all the above attributes to route each flow on one of its eligible paths. When solving the incremental problem, the algorithm can only make use of dynamic attributes.

B. *Step 1: Ordering of Offered Flows*

Given the list of flows to be routed, one can arbitrarily select the next flow to be routed. This arbitrary selection is the only option when solving the incremental problem as it there are no dynamic flow attributes. However, when solving the static problem, our results have shown that taking the above flow attributes into account has a considerable effect on bringing the routing results closer to the optimal ILP solution.

Therefore, only when solving the static problem, our heuristic computes the values of the flow attributes for each flow on the flow list and then sorts the list of flows in a way that is computationally efficient while boosting the likelihood of sharing common light-trails among routed flows.

Considering the demand (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. The reason is that already occupied light-trails are less likely to have enough capacity to accommodate such flows with high demand value. Therefore, we start routing the flows with higher demand first by sorting the list of flows in the descending order of D_f and then select the flow on top of the list.

Considering the number of eligible paths (EP_f), we note that this attribute represents the allowed degree of freedom when making the routing decision of a flow. It is thus preferable to begin routing the 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. Therefore, we sort the list of flows in the ascending order of EP_f and then select the flow on top of the list.

To avoid sorting the list of flows with multiple attributes; we reduce the computational complexity of the sorting function by combining both flow attributes, using a weighted sum, in one *flow-preference* attribute (Q_f) as follows:

$$Q_f = W_D \times D_f - W_{EP} \times EP_f \tag{11}$$

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. Deciding which attribute to be the primary sorting key is critical to the quality of the obtained routing solution. We propose a simple rule for making this decision by first computing:

$$EP_{\min} = \min_{f \in F} EP_f \tag{12}$$

We choose EP_f to be the primary sorting key of the flow list if $EP_{\min}$ is less than *MinNumLTs* given by (1). The reasoning behind this rule is that if $EP_{\min}$ is higher than the minimum number of routes used to carry the offered flows; then most flows should have a sufficient number of eligible paths to be routed on. The effect of the EP_f attribute is thus less significant than the D_f attribute which should then be used as the primary sorting key.

To have a certain attribute to be the primary sorting key, its weight must be higher than the maximum difference between any two values of other attributes with their weights normalized to unity.

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

$$W_{EP} = 1,\ W_D = \Delta\ EP_{\max} + 1 \qquad (13)$$

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:

$$W_D = 1,\ W_{EP} = \Delta\ D_{\max} + 1 \qquad (14)$$

Where:

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

C. Step 2: Routing of Ordered Flows

Having selected the flow (f) to be routed next; the algorithm proceeds to select an eligible path (p) with enough spare capacity. As with the flow ordering process, the list of eligible paths is sorted using the path attributes. The paths should be selected in a way that is both computationally efficient and is favorable of routing more flows on highly-shared paths rather than using separate underutilized paths.

Considering a path p, a higher value of ED_p and EF_p 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_p and RF_p indicates that the path has been already used to route more past flows; again implying more preference for selecting it to boost capacity sharing.

To avoid sorting with multiple attributes, we combine the path attributes using a weighted sum in one path-preference Q_p attribute as follows:

For the static problem:

$$Q_p = W_{RD} \times RD_p + W_{RF} \times RF_p + W_{ED} \times ED_p + W_{EF} \times EF_p \qquad (15)$$

For the incremental problem:

$$Q_p = W_{RD} \times RD_p + W_{RF} \times RF_p \qquad (16)$$

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.

Setting the values of weights W_{RD}, W_{RF}, W_{ED}, and W_{EF} allows the algorithm to tune which attribute is most prevalent in making the routing decision. Our results have shown no significant effect of the attributes that are based on the number of flows (EF_p and RF_p). Using a value of unity for the W_{EF} and W_{RF} had no effect on the obtained results. This is consistent with the fact the actual network capacity is limited in terms of the channel bandwidth rather than the number of flows per channel. We are thus left with the attributes RD_p and ED_p.

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 (RD_p) is still zero for most paths. Using random tie breaks for those initial routing decisions greatly degrades the quality of the final routing results as the algorithm tends to pack all remaining flows on the initial randomly selected routes. It is therefore critical to make use of the amount of eligible traffic attribute (ED_p) as a guiding parameter for the initial routing decisions.

Our algorithm adopts to the former consideration by setting the weights (W_{ED} and W_{RD}) in a way that allows 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. Therefore, the weights are calculated as:

$$W_{RF} = W_{EF} = W_{ED} = 1 \qquad (17)$$

$$W_{RD} = \Delta\ ED_{\max} + 1 \qquad (18)$$

Where:

$$\Delta ED_{\max} = ED_{\max} - ED_{\min},\ ED_{\max} = \max_{p \in P} ED_p\ ,\ ED_{\min} = \min_{p \in P} ED_p$$

It is worth noting that the value of Q_p has the potential to be the same for multiple paths when selecting the path to route the current flow. This is especially more probable for the initial flows when solving the incremental problem where Q_p is typically zero in (16). In such cases, our heuristic randomly breaks the tie and arbitrarily selects a path.

D. Complexity Analysis

Proposition: The running time complexity of the algorithm described by Fig. 5 is $O(N^2 d^{H-1} log\ d)$ where d is the maximum connectivity degree of a network node and H is the limit on the hop count of DFS discovered paths.

Proof: The heuristic starts by sorting the list of flows according to the calculated flow-preference (Q_f) given by (11). The Maximum number of unidirectional flows in any problem is $N(N\text{-}1)$; that is $O(N^2)$. Thus, the first sorting operation is $O(N^2 log\ N)$.

Next, the algorithm loops over all flows to route them one-by-one; so this loop is executed $O(N^2)$ times. In each iteration of this loop; the algorithm sorts the eligible paths per flow according to the calculated path-preference (Q_p) given by (15) (or (16) for the incremental traffic case). This sorting operation have a running time of $O(EP_f\ log\ EP_f)$. Therefore, the running time of the proposed heuristic is:

$$O(N^2 log\ N) + O(N^2\ EP_f\ log\ EP_f) \qquad (19)$$

Solving the incremental problem does not involve sorting of the flow list and hence its order of complexity only includes the second term of (19).

It then needed to derive $O(EP_f)$ and then substitute in (19). Let d be the maximum node degree over all network nodes; let h be the hop count of a path. We prove that EP_f is $O(H^3 d^{H-1})$. The derivation follows by considering Fig. 6 which shows the number of eligible paths passing by the source (S) and destination (T) nodes of a flow for the case of $h = 3$.

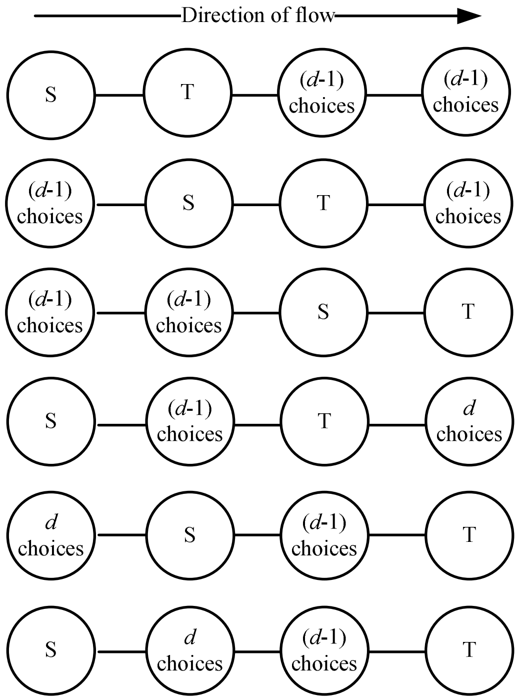

Fig. 6: Computing the number of eligible paths per flow

We see that the number of eligible paths for a flow from node S to node T is equal to 3 $(d\text{-}1)^2$ + 3 $d(d\text{-}1)$ which is $O(6d^2)$. We generalize this to the case of any value of h by considering that each row of Fig. 6 corresponds to a different position of S and T nodes. The number of such rows for a path of h hops is computed as follows: Among (h+1) nodes; node S can take one of h positions ranging from top upstream to penultimate downstream position. When S is the top upstream node; there can be h possible positions for T and when S is the next to top upstream node; there can be h-1 positions for T and so on until S is in the penultimate downstream position where T can only take the single position of the most downstream node. Thus, similar to Fig. 6, the number of rows for h hops:

$$h + (h-1) + \ldots + 2 + 1 \Rightarrow (h + 1)\, h/2 \qquad (20)$$

For each specific placement of S and T nodes, the remaining number of nodes is h-1 nodes. Each node can be selected in $O(d)$ times; therefore there is $O(d^{h-1})$ paths for each specific placement of S and D nodes which corresponds to one row of Fig. 6. Given (20), we conclude that the number of eligible paths of a hop count of h for a flow f is $O(h^2 d^{h\text{-}1})$. Therefore, for a hop limit of H:

$$O(EP_f) \equiv O(\sum_{h=1}^{H} h^2 d^{\,h-1}) \equiv O(H^3 d^{H\text{-}1}) \qquad (21)$$

Combining (19) and (21), the worst-case running time of the proposed heuristic for the static traffic case is:

$$O(N^2 log\ N) + O(N^2 H^3 d^{H\text{-}1} log\ (H^3 d^{H\text{-}1})) \qquad (22)$$

Hop count (H) is typically a limited constant; the second term of (22) can be reduced as:

$$O(N^2 H^3 d^{H\text{-}1} log\ (H^3 d^{H\text{-}1})) \equiv O(N^2 d^{\,H\text{-}1} log\ d) \qquad (23)$$

Substituting (23) in (22); the worst-case running time for the static problem is:

$$O(N^2 log\ N) + O(N^2 d^{\,H\text{-}1} log\ d) \qquad (24)$$

Noting that $d^{\,H\text{-}1} log\ d > log\ N$, for typical network topologies, only the second term of (24) dominates. Furthermore, the complexity of the incremental problem only involves the second term of (24) which completes the proof.

VI. Results

In this section, we solve three problems using both ILP and heuristic techniques. As discussed in section V-C, the results of both incremental and static problems is generally dependent on the initial ordering of offered flows and discovered network paths due to the possibility of making some routing decisions based on random tie breaks. This fact is more pronounced when solving the incremental problem in which only the dynamic attributes counts to the value of Q_p.

To assess that our results are not specific to a particular initial ordering of those lists, we solve each problem 10 times with a different randomized order for each run. We report the result parameters in terms of their average values and the percentage ratio of standard deviation to the average value.

We report the following parameters for a routing solution:

1- Number of signaled light-trails to carry the offered traffic,
2- Number of signaled light-trails carrying 95% of the amount of offered traffic, and
3- The total number of wavelength links used by the signaled light trails.

Many other parameters may be reported using our code such the maximum number of used wavelengths; we do not include them here due to lack of space.

A. Problem 1: A 10-node Mesh Network

Fig. 7 and Table II give the network topology and traffic matrix used in [6] respectively. We assume a hop limit of 4 hops when discovering network paths using DFS.

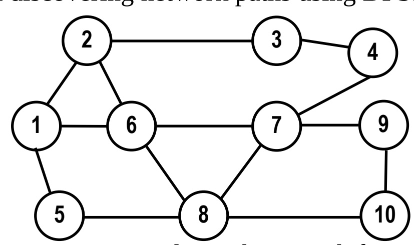

Fig. 7: A 10-node mesh network from [6]

TABLE II: Traffic matrix of the network of Fig. 7

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

The optimal solution routes the offered traffic over 13 light-trails [6]. Using the proposed heuristic, the total traffic was routed over 20 light-trails for the static traffic case. However, more than 94% of the offered traffic was routed over the same optimal number of 13 light-trails. Table III gives the routing result of the static problem.

TABLE III: STATIC TRAFFIC ROUTING RESULT USING THE PROPOSED HEURISTIC ON THE NETWORK OF FIG. 7

No.	Path	Carried Flows	Load
1	(5,1,6,7,9)	(1,6)(1,7)(5,6)(5,7)(6,9)(1,9)(5,9)	48
2	(5,1,6,8,10)	(5,1)(8,10)(6,10)(1,10)(5,10)	48
3	(9,7,6,2,3)	(7,2)(2,3)(6,3)(9,6)(9,2)(9,3)	47
4	(9,10,8,5,1)	(8,5)(9,8)(9,10)(10,1)(9,1)(10,5)(9,5)	46
5	(9,10,8,6,7)	(8,6)(8,7)(10,8)(9,7)(10,7)(10,6)	46
6	(10,8,7,4,3)	(4,3)(7,3)(8,4)(8,3)(10,3)	46
7	(3,4,7,9,10)	(7,10)(3,4)(3,7)(4,10)(4,9)(3,9)(3,10)	46
8	(2,6,8,7,9)	(6,7)(2,6)(7,9)(2,8)(2,7)(8,9)(2,9)	46
9	(3,4,7,6,8)	(6,8)(7,6)(7,8)(4,7)(3,6)(4,8)(3,8)	45
10	(4,7,6,1,5)	(1,5)(7,1)(6,5)(7,5)(4,6)(4,1)(4,5)	43
11	(5,1,6,2,3)	(6,2)(1,2)(5,2)(1,3)(5,3)	39
12	(5,1,6,7,4)	(7,4)(6,4)(1,4)(5,4)	30
13	(2,1,6,8,10)	(1,8)(2,1)(2,10)	16
14	(2,6,8,5,1)	(6,1)(8,1)(2,5)	15
15	(2,3,4)	(2,4)	5
16	(10,9)	(10,9)	4
17	(3,4,7,6,1)	(3,1)	3
18	(4,7,6,2)	(4,2)	2
19	(5,8)	(5,8)	2
20	(9,10,8,7,4)	(9,4)	2

Manual inspection of the above results could reveal that the routing decisions for some flows did not serve the purpose of the algorithm. As an example, flows (3,1) and (4,2) could have been routed on a single light-trail (4,3,2,1). However, this is an expected “glitch” of the algorithm operation due to the initial routing decisions endeavoring to increase the likelihood of more flows sharing the same light-trails. The decision is only based on the amount of eligible traffic of such paths, noting that, initially, the amount of routed traffic on all paths is typically zero.

TABLE and Fig. 8 show the routing result parameters of the three solutions (optimal, static, and incremental).

TABLE IV: PARAMETERS OF ROUTING RESULTS OF THE NETWORK OF FIG. 7

Problem/ Parameter Value as “avg. (%std. dev.)”	No. of used LTs	No. of LTs carrying 95% of traffic	No. of used wavelength links
Optimal	13	13	52
Static (Table III)	20 (0%)	14 (0%)	71 (0%)
Incremental	26.9 (6.8%)	21.3 (7%)	91.7 (4.2%)

B. Problem 2: NSFNET Model

Fig. 8 shows the 14-node NSFNET network given in [8]. We assume a hop limit of four hops.

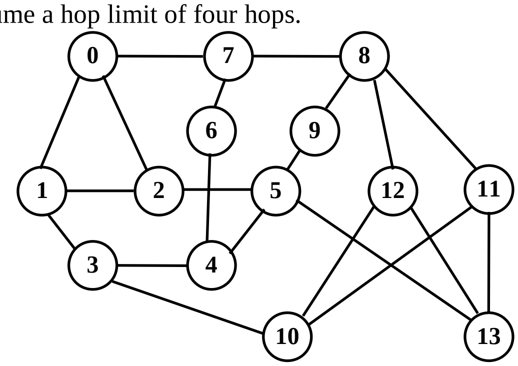

Fig. 8: NSFNET Topology

We solve this problem two times with two different traffic matrices with varying demand intensities derived from [8]. On solving the ILP problem for both cases; GLPK could not find the optimal integer solution in a reasonable running time. Therefore, we had to use the achieved feasible solution after a sufficiently high number of iterations (20 millions).

We use a similar idea as in [8] to generate the first traffic matrix, Table V, which is based on randomly selecting 42 flows with a demand value uniformly distributed over (0, 48) while allocating a demand value uniformly distributed over (0,2) to other flows. This captures a situation where most of the network traffic is concentrated among 42 pairs, with little traffic among the remaining ones. The achieved ILP solution used 30 light-trails. The static traffic heuristic was able to use an average of 44.7 light-trails over 10 runs. TABLE and Fig. 8 shows the routing result parameters of the three solutions (optimal, static, and incremental)

TABLE V: TRAFFIC MATRIX M1 OF THE NETWORK OF FIG. 8

From/To	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	4	0	25	1	42	1	2	0	1	1	0	0	2
2	1	0	0	1	2	28	0	9	1	0	39	0	0	1
3	34	0	0	0	2	2	6	15	27	2	2	0	0	1
4	1	2	1	0	2	18	0	1	0	44	1	2	1	2
5	1	0	2	40	0	0	40	1	0	1	1	1	2	1
6	0	2	2	2	1	0	0	2	1	2	1	13	40	31
7	1	0	0	2	36	0	0	19	20	1	1	10	1	1
8	18	2	0	2	0	2	0	0	29	1	2	1	37	1
9	0	23	1	24	0	1	31	3	0	28	0	1	2	0
10	25	1	2	1	34	0	1	3	2	0	2	0	2	0
11	1	1	45	0	0	0	1	38	0	0	0	1	0	1
12	0	0	46	1	2	1	1	1	2	1	2	0	0	39
13	2	1	0	1	2	2	2	2	10	0	36	37	0	2
14	1	46	1	0	1	0	1	0	1	2	0	26	0	0

TABLE VI: PARAMETERS OF ROUTING RESULTS OF TRAFFIC MATRIX M1 OVER THE NETWORK OF FIG. 8

Problem/ Parameter Value as “avg. (%std. dev.)”	No. of used LTs	No. of LTs carrying 95% of traffic	No. of used wavelength links
ILP (non-optimal)	30	28	117
Static	44.7 (2.1%)	28.8 (1.5%)	161.6 (1.6%)
Incremental	49.6 (6.5%)	31.3 (2.6%)	166.1 (4.7%)

The second traffic matrix, Table VII, corresponds to a measured traffic distribution taken from [9] with traffic distributed more evenly over a large number of source-destination pairs. The achieved ILP solution used 33 light-trails. Solving the static problem using the proposed heuristic; we were able to route the offered traffic over an average of 40.7 light-trails. Table and Fig. 8 show the routing result parameters of the three solutions (optimal, static, and incremental)

Table VII: TRAFFIC MATRIX M2 OF THE NETWORK OF FIG. 8

From/To	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	2	2	0	0	0	0	2	2	0	0	2	0	0
2	6	0	6	2	6	4	2	8	2	8	2	18	4	6
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	2	8	0	2	2	2	2	2	2	2	4	0	2
5	0	32	10	2	0	2	6	30	12	10	0	38	6	4
6	0	4	2	4	2	0	2	2	2	2	0	2	0	2
7	2	12	48	2	12	4	0	48	10	12	2	34	0	16
8	2	30	48	6	12	2	46	0	22	16	6	24	4	8
9	4	10	18	4	12	4	12	28	0	18	8	48	12	2
10	0	16	6	2	12	6	4	28	4	0	4	10	4	2
11	0	2	2	2	0	0	2	2	2	4	0	2	0	2
12	2	12	2	4	12	2	4	20	14	12	6	0	2	2
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	2	14	8	6	18	6	4	14	4	0	2	0	6	0

Table VIII: PARAMETERS OF ROUTING RESULTS OF TRAFFIC MATRIX M2 OVER THE NETWORK OF FIG. 8

Problem/ Parameter Value as "avg. (%std. dev.)"	No. of used LTs	No. of LTs carrying 95% of traffic	No. of used wavelength links
ILP (non-optimal)	33	30	117
Static	40.7 (1.2%)	28.7 (1.7%)	141.3 (1.3%)
Incremental	50.1 (6.3%)	35.7 (6.3%)	160 (6.2%)

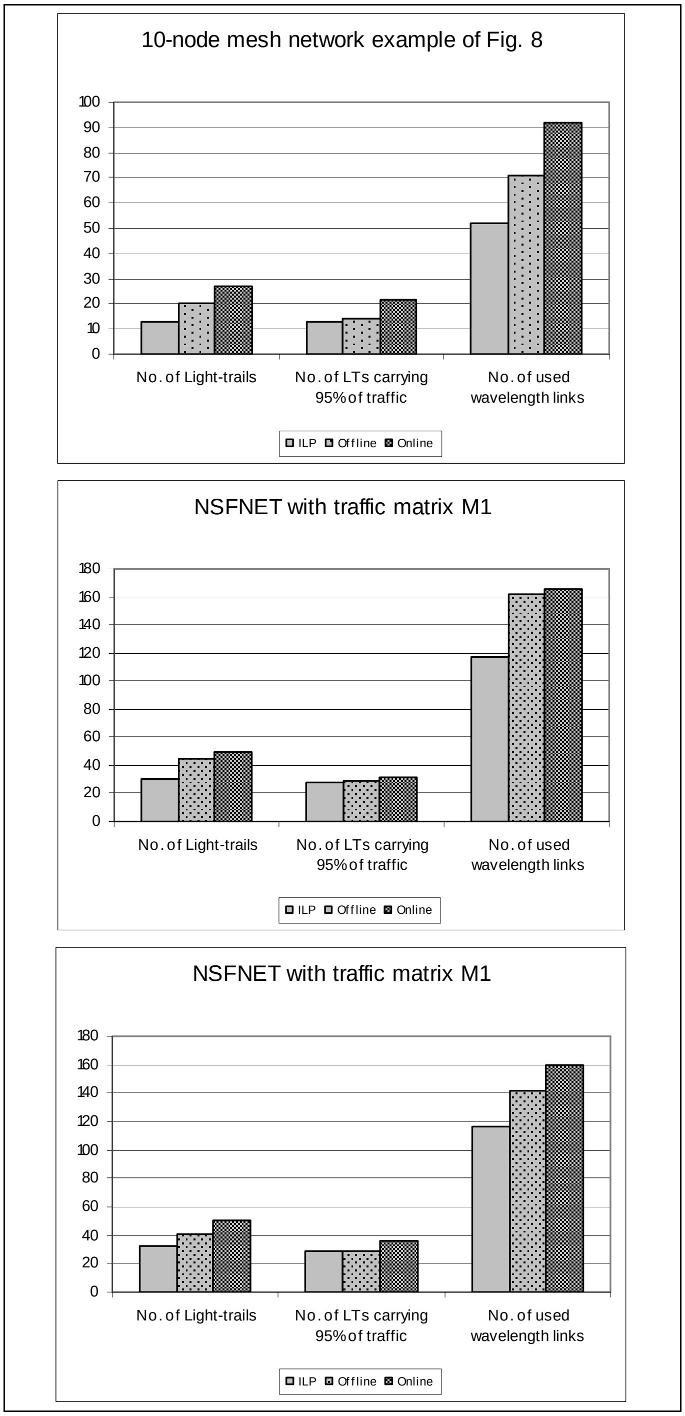

Fig. 8: Summary of results of the three solved problems

VII. CONCLUSION AND FUTURE WORK

Light-trails provide a technologically feasible alternative for the efficient use of DWDM channels when transporting IP traffic. Proper routing of traffic flows is essential to the establishment of light-trails carrying more flows per DWDM channel and thus achieving higher utilization.

We have proposed two enhancements to reducing the number of constraints and narrowing the search space of the ILP formulation of the 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. The worst-case running time complexity of the proposed heuristic was derived.

Our heuristic gives near-optimal results in terms of the number of light-trails used to carry the offered traffic. It has also been able to pack more 95% of the offered traffic within a number of light-trails that is very close to the optimal minimum. We have also applied our heuristic to the case of incremental problem where we were able to only make use of the dynamic path attributes when selecting the best path for the new flow. Incremental traffic solutions were further from optimal and static traffic results as expected.

Our future work involves porting the proposed heuristic to the survivable case where each flow is assigned to two link-disjoint paths for backup purposes against link failures.

REFERENCES

[1] A. Gumaste, "Light-Trails and light-frame architectures for optical Networks," Ph.D. thesis, Fall 2003, The University of Texas at Dallas.

[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 Arun K. Somani, "Optimal Light Trail Design in WDM Optical Networks," in proceedings of ICC 2004, Vol. 3, June 2004, pp. 1699-1703

[7] Andrew Makhorin, GNU Linear Programming Kit (GLPK), Department for Applied Informatics, Moscow Aviation Institute, Moscow, Russia.

[8] R. Ramaswami and K. N. Sivarajan, "Design of logical topologies for wavelength-routed optical networks," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 5, pp. 840–851, June 1996.

[9] B. Mukherjee, S. Ramamurthy, D. Banerjee, and A. Mukhrejee, "Some principles for designing a wide-area optical network," in *Proc. IEEE INFOCPM '94, 1994*.

[10] A Gumaste and I. Chlamtac, 'Mesh Implementation of Light-trails,' IEEE Proceedings of International Conference on Communication (12th) ICCCN Dallas, October 20. 2003.

[11] Yabin Ye, Hagen Woesner, Roberto Grasso, Tao Chen, Imrich Chlamtac, 'Traffic Grooming in Light Trail Networks,' in *Proc. IEEE GLOBECOM 2005*.

[12] Srivatsan Balasubramanian, Arun K. Somani, "Traffic Grooming in Statistically Shared Optical Networks," IEEE LCN 2006

[13] A. Gumaste, P. Palacharla and S. Kinoshita, "Light-trail based Shared Wavelength Optical Network," 10th Opto-Electronics Communication Conference (OECC) 2005 Korea

[14] A. Gumaste, P. Palacharla and T. Naito, "Experimental Demonstration for Static and Dynamic Control for Light-trail Networks," 11th Opto-Electronic Communications Conference OECC 2006, Taiwan.

[15] S. Balasubramanian, A.E. Kamal, A.K.Somani, "Network Design for IP-Centric light-trail networks," IEEE Broadnets 2005.