Light-Trail Assignment in WDM Optical Networks

Bin Wu and Kwan L. Yeung
Dept. of Electrical and Electronic Engineering
The University of Hong Kong
Pokfulam, Hong Kong
E-mail: {binwu, kyeung}@eee.hku.hk

***Abstract*—Light-trail has emerged as a promising candidate for enabling IP over WDM networks. The problem of static light-trail assignment is to find a set of light-trails to cover the given traffic demands, such that the total number of light-trails required is minimized. Because of the power loss caused by splitting at each hop, the length of a light-trail is limited. Existing light-trail assignment algorithms adopt ILP (Integer Linear Programming) approach. Due to the high complexity of ILP, such algorithms are not scalable. In this paper, we propose an efficient heuristic algorithm LTA (Light-Trail Assignment) to solve this problem. In LTA, each light-trail is judiciously assigned based on the *request discreteness*, the *shortest path length* and the *traffic volume* of each request. A reference node mechanism is also designed to enhance the solution. Numerical results show that LTA always returns sub-optimal solutions.**

Keywords-Heuristic; integer linear programming (ILP); light-trail assignment; wavelength division multiplexing (WDM).

I. INTRODUCTION

IP over WDM is an appealing architecture for meeting the ever-increasing data demands in the Internet. The bandwidth of a WDM wavelength is usually 10Gbps, whereas the bandwidth required by sub-rate traffic varies from STS-1 (51.84 Mbps) to the full wavelength capacity. In traditional optical flow/circuit switching (OFS) networks, an all-optical *lightpath* is set up between two communicating nodes using a dedicated wavelength. When the traffic load is light, the wavelength capacity is under-utilized. To improve the wavelength utilization, optical packet switching (OPS) [1] and optical burst switching (OBS) [2] are proposed. However, such techniques require either expensive or immature optical components such as optical buffers and fast optical switching.

Recently, *light-trail* [3] has been proposed as a promising technique for IP over WDM. A light-trail is a unidirectional optical bus between a convener node (i.e. light-trail head) and an end node. It is implemented using a wavelength, and allows the intermediate nodes to share the bandwidth by adding or dropping traffic, provided that the total traffic load carried is not more than the bandwidth of the wavelength. An upstream node can send data to any downstream node, and a downstream node only accepts those packets destined to it. At each intermediate node, this is achieved by splitting a percentage of optical power and detecting the optical signals using a transponder. Meanwhile, the rest of the optical signals continue to propagate along the light-trail. A signaling mechanism is used to ensure conflict-free communications among the nodes [3]. Such a light-trail architecture is based on mature optical technologies. It improves wavelength utilization without using optical buffers, fast optical switching and O-E-O conversions at the intermediate nodes. Due to the power loss caused by splitting at each hop, the length of a light-trail is limited and is expected to be not more than 5 hops [4].

Light-trail attracts a lot of research interests [3-11]. The node architecture and signaling mechanisms are detailed in [3-4]. Performance evaluations can be found in [3-5]. ILP formulations are given in [6-7] for light-trail assignment. Traffic grooming [9], dynamic routing and protection [10] are also studied in light-trail networks. Besides, bidirectional light-trail (BDLT) is recently proposed to enhance the fairness performance [11].

In this paper, we propose an efficient heuristic LTA (Light-Trail Assignment) to assign light-trails under a given traffic matrix. To the best of our knowledge, LTA is the first heuristic designed so far. Compared to the ILP approach, LTA is more scalable and is shown to always return a sub-optimal solution. The rest of the paper is organized as follows. In Section II, the light-trail assignment problem is formulated. LTA is proposed in Section III. Section IV shows the numerical results and Section V gives some discussion. We conclude the paper in Section VI.

II. LIGHT-TRAIL ASSIGNMENT PROBLEM

For a network with N nodes, an $N\times N$ traffic matrix $\boldsymbol{T}=\{t_{ij}\}$ is used to denote the traffic demands among all node pairs. Let C be the capacity of a wavelength and assume each individual demand/request $t_{ij}\le C$. The light-trail assignment problem is to determine a minimum set of light-trails to carry all the requests in $\boldsymbol{T}$. For each light-trail, its length (measured in hops) must not exceed a predefined value $L_{\max}$, and the total traffic carried must not exceed C.

Let h_{ij} be the length of the shortest path between nodes i and j. If $h_{ij}\le L_{\max}$, flow t_{ij} is carried by a single light-trail. If $h_{ij}>L_{\max}$, more than one concatenated light-trails are required. In the latter case, each light-trail end/head node introduces an O-E-O conversion. So it is necessary to minimize the number of concatenated light-trails required. While this can be achieved in many ways, we reuse the *traffic matrix preprocessing* in [6], which is reproduced in Fig. 1. The idea is to greedily break the shortest path from node i to node j into several segments with length $L_{\max}$ each, and the length of the

```
Traffic Matrix Preprocessing
While (find $(i, j) : t_{ij} > 0, h_{ij} > L_{\max}$ )
{
    1. Pick an intermediate node k:
            $k = \arg\min_{v\in V} \{ h_{vj} \mid h_{iv} \le L_{\max} \}$;
    2. Update traffix matrix $\boldsymbol{T}$:
            (a) $t_{ik} \leftarrow t_{ik} + t_{ij}$ ;
            (b) $t_{kj} \leftarrow t_{kj} + t_{ij}$ ;
            (c) $t_{ij} \leftarrow 0$ ;
}
```

Fig. 1. Traffic matrix preprocessing (V is the set of all the nodes).

This work is supported by Hong Kong Research Grant Council Earmarked Grant HKU 7150/04E.

last segment can be smaller than $L_{\max}$. Accordingly, the value t_{ij} is added to the existing traffic volume between the two end nodes of each segment.

Two key issues are involved in the light-trail assignment problem: determining the set of requests to be carried by each light-trail, and minimizing the total number of light-trails required. Light-trail assignment is similar to the *knapsack* and *bin packing* problems, which are both NP-hard. It differs from them in having the extra light-trail length constraint ($L_{\max}$). This makes it even harder to solve [6].

Due to the intractability of the problem, existing algorithms [6-7] rely on ILP. In [6], all the paths with hop-count $L_{\max}$ are first found by *breath first search*. They form a set of candidate trails $\boldsymbol{L_T}$. Then, a subset of $\boldsymbol{L_T}$ is returned by ILP as the light-trails. However, the ILP approach is not scalable.

III. LTA Algorithm

A. Observations

To minimize the required number of light-trails, intuitively we can design a greedy algorithm to let a light-trail carry as much traffic as it can. But this is not enough as illustrated in Fig. 2. Assume $L_{\max}$=3 and C=48. The number next to each dashed arrow in Fig. 2 gives the value of t_{ij}. In Fig. 2a, if we first use two light-trails 0→1→2→3 and 4→5→6→7 to carry the requests $\{t_{01}, t_{12}, t_{23}\}$ and $\{t_{45}, t_{56}, t_{67}\}$ in a greedy manner, then the remaining requests t_{34} and t_{70} need two additional light-trails. This is because the length of a light-trail must be no more than $L_{\max}$=3 hops, and t_{34}, t_{70} are not close enough to share a single light-trail. However, if light-trails 0→1→2→3 and 3→4→5→6 are used to carry $\{t_{01}, t_{12}, t_{23}\}$ and $\{t_{34}, t_{45}, t_{56}\}$, then the remaining requests t_{67} and t_{70} can share a common light-trail. This gives a better solution of three light-trails only. This example shows the importance of *request discreteness*.

In Fig. 2b, if we first pack $\{t_{01}, t_{12}, t_{23}\}$ onto light-trail 0→1→2→3, then the remaining requests t_{03} and t_{42} require two additional light-trails. Note that h_{03} and h_{42} are larger than h_{ij} of the other three requests. If we first pack the longest request t_{03} with t_{12} and t_{23} onto light-trail 0→1→2→3, then the request t_{01} can take a detour to share light-trail 0→4→1→2 with t_{42}. The latter case requires one less light-trail than the former. This example shows that *it is better to consider requests t_{ij} with large h_{ij} first, because a request with smaller h_{ij} usually has more flexibility in sharing a common light-trail with others.*

In Fig. 2c, we first consider t_{02} because h_{02} is the largest. However, if t_{02} and t_{23} are packed onto light-trail 0→1→2→3, then the remaining bandwidth of this light-trail (48−28=20) is not sufficient to carry t_{01} (=25). As a result, t_{40} and t_{01} need two additional light-trails. However, if t_{02} is packed with t_{40} or t_{01}, the remaining requests can share a common light-trail. This example shows that *we should first accommodate those requests with large t_{ij}, as it is generally easier for a request with smaller t_{ij} to share a common light-trail with others.*

We argue that the importance of the above three factors are ranked in the same order as they were discussed. If some requests are far from each other, it is impossible for them to share a common light-trail regardless of the values of h_{ij} and t_{ij}. So, request discreteness has the largest impact. On the other hand, because of the hop constraint $L_{\max}$, h_{ij} has a larger impact than t_{ij}. For example, in Fig. 2b, no matter what are the values of t_{03} and t_{42}, it is impossible for them to share a light-trail of $L_{\max}$=3. We can consider the flow volume t_{ij} only when it is possible to pack the requests onto a common light-trail.

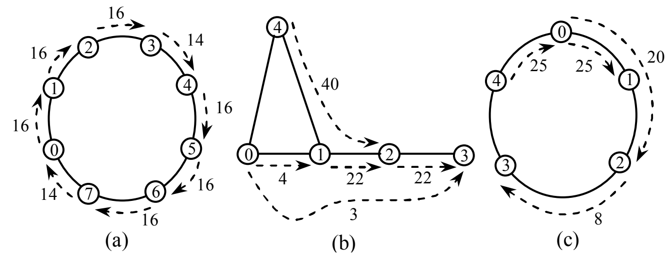


Fig. 2. Three examples with $L_{\max}$=3 and C=48.

B. General Idea of LTA

We first introduce the notion of *reference node*. A reference node r is a node based on which we can calculate how far a request t_{ij} is away from it. The *distance* between t_{ij} and r is defined as

$$d_{ij}^{r} = h_{ir} + h_{jr} \,. \tag{1}$$

In LTA, we first consider the request with the largest d_{ij}^{r}. If a request t_{ij} is carried by a light-trail, we set its d_{ij}^{r} to 0. As the number of determined light-trails increases, the maximum value of d_{ij}^{r} decreases, and the not-yet-carried requests gradually converge to the reference node. This helps to address the request discreteness issue. If multiple requests have the same d_{ij}^{r}, we consider the one with the largest h_{ij} first. Similarly, if multiple requests have the same d_{ij}^{r} and h_{ij}, the one with the largest traffic volume t_{ij} is considered first.

Assume $\boldsymbol{L_T}$ is the candidate trail set found by breadth first search, and t_{ab} is the farthest request from r (or *the most crucial request*). We find a subset of trails $\boldsymbol{L_S} \subseteq \boldsymbol{L_T}$ that can carry t_{ab}. For each trail $l \in \boldsymbol{L_S}$, we invoke a packing process *pack*(l) (to be discussed later). It returns a *tentative packing scheme* $\boldsymbol{P(l)}$ which contains all the requests (denoted by node pairs) that can be carried by l. Then, the best one l_{best} is determined from (2) as a light-trail, and the packing scheme for l_{best} is stored in $\boldsymbol{P(l_{best})}$.

$$l_{best} = \arg \max_{l\in \boldsymbol{L_S}} \{sum\_t\}\Big|_{\max_{l\in \boldsymbol{L_S}}\{sum\_h\}} \tag{2}$$

where

$$sum\_h = \sum_{(i,j)\in \boldsymbol{P(l)}} h_{ij} \,, \tag{3}$$

$$sum\_t = \sum_{(i,j)\in \boldsymbol{P(l)}} t_{ij} \,. \tag{4}$$

From (2)~(4), the trail with the maximum *sum_h* is chosen as l_{best}. In case of multiple choices, the one with the heaviest traffic load (or the maximum *sum_t*) is chosen. Then, both t_{ij} and h_{ij} are set to 0 for $(i, j) \in \boldsymbol{P(l_{best})}$. We repeat this process until all $t_{ij} \in \boldsymbol{T}$ are carried.

However, it is difficult to determine the best reference node r. A simple technique is to try out every node as r and get a solution $\boldsymbol{Sol_r}$. Then the best solution is chosen among $\{\boldsymbol{Sol_r} \mid 0 \le r < N\}$. This simplifies the problem, and provides a way for N solutions. The cost is an extra factor of N in time complexity.

C. Packing Process

We apply the packing process *pack*(l) to each $l \in \boldsymbol{L_S}$. The total possible number of requests that can be packed onto l is

```
Packing Process pack(l)
Input:
    The most crucial request $t_{ab}$; a trail $l$ that can carry $t_{ab}$; updated traffic
    matrix $T=\{t_{ij}\}$; updated all-pairs shortest distance matrix $H=\{h_{ij}\}$;
    updated reference matrix $D^r=\{d_{ij}^r\}$; light-trail capacity $C$.
Output:
    Tentative packing scheme $P(l)$ for trail $l$; $sum\_h$ and $sum\_t$.

1. Initialization:
    $\Phi \to P(l)$; $\Phi \to \Delta$; Let $l(n)$, $L_{max} \ge n \ge 0$ be the $n$-th node along $l$.
2. Find all node pairs (s,d) covered by l:
    for ( $i=0$; $i<L_{max}$; $i$++ )
        for ($j=0$; $j<i+1$; $j$++ )
        {
            $s=l(j)$;
            $d=l(L_{max}-i+j)$;
            if ($t_{sd} \ne 0$) $\{(s,d)\} \cup \Delta \to \Delta$;
        }
3. Pack the most crucial request $t_{ab}$:
    $h_{ab} \to sum\_h$; $t_{ab} \to sum\_t$;
    $\{(a,b)\} \cup P(l) \to P(l)$; $\Delta - \{(a,b)\} \to \Delta$.
4. Pack adjacent requests:
    while ($\Delta \ne \Phi$)
    {
        $max\_h = \{(s,d) | (s,d) = \arg\max_{(s,d)\in\Delta}\{h_{sd}\}\}$;
        $max\_hd = \{(s,d) | (s,d) = \arg\max_{(s,d)\in max\_h}\{d_{sd}^r\}\}$;
        $max\_hdt = \{(s,d) | (s,d) = \arg\max_{(s,d)\in max\_hd}\{t_{sd}\}\}$;
        while ( $\exists (s,d) \in max\_hdt$ )
        {
            $\Delta - \{(s,d)\} \to \Delta$;
            $max\_hdt - \{(s,d)\} \to max\_hdt$;
            if ($sum\_t + t_{sd} > C$) continue;
            else
            {
                $\{(s,d)\} \cup P(l) \to P(l)$;
                $sum\_h + h_{sd} \to sum\_h$;
                $sum\_t + t_{sd} \to sum\_t$;
            }
        }
    }
5. Results:
    Return $P(l)$, $sum\_h$ and $sum\_t$.
```

Fig. 3. Packing process *pack*(l). The inputs $\boldsymbol{T}$, $\boldsymbol{H}$ and $\boldsymbol{D}^r$ are updated in Fig. 4.

```
LTA ALGORITHM
Input:
    a) Light-trail length limit $L_{max}$ and wavelength capacity $C$;
    b) A network with nodes numbered from 0 to $N-1$;
    c) A traffic matrix $T=\{ t_{ij} | t_{ij} \le C\}$.
Output:
    A solution $Sol$ containing a minimum set of light-trails, with a
    packing scheme for each light-trail.

Step 1) Initialization:
    Use Floyd-Warshall algorithm to calculate the all-pairs shortest
    distance matrix $H=\{h_{ij}\}$. Apply the traffic matrix preprocessing in Fig. 1
    to $T$. Find all candidate trails with hop-count $L_{max}$ by breath first search
    and record them in the candidate trail set $L_T$. Initialize $Sol_0$, …, $Sol_{N-1}$ to
    null. Set $0 \to r$, $T \to T_{bak}$ and $H \to H_{bak}$.
Step 2) Construct $Sol_r$:
    a) Based on the reference node $r$ and formula (1), calculate the
    distance matrix $D^r=\{d_{ij}^r\}$ from $H$. If $t_{ij}=0$ then set $0 \to d_{ij}^r$.
    b) Find the most crucial request $t_{ab}$ with the largest $d_{ij}^r$. In case of
    multiple choices, consider the request with the largest $h_{ij}$ first. If multiple
    requests have the same values of $d_{ij}^r$ and $h_{ij}$, choose the one with the
    largest $t_{ij}$.
    c) Find a subset of trails $L_S \subseteq L_T$ that can carry $t_{ab}$.
    d) Apply the packing process pack(l) in Fig. 3 to each trail $l \in L_S$.
    Determine the best trail $l_{best}$ as a light-trail according to (2). Record $l_{best}$
    and the corresponding packing scheme $P(l_{best})$ to $Sol_r$.
    e) For each node pair $(i,j) \in P(l_{best})$, set $0 \to t_{ij}$, $0 \to h_{ij}$ and $0 \to d_{ij}^r$.
    f) If $\exists (t_{ij} \in T, t_{ij} \ne 0)$, go to Step 2b). Otherwise continue.
    g) Set $T_{bak} \to T$, $H_{bak} \to H$ and $r+1 \to r$. If $r<N$, go to Step 2a).
    Otherwise go to Step 3).
Step 3) Determine the final solution Sol:
    Among the $N$ solutions $\{Sol_r | 0 \le r < N\}$ obtained, choose the one with
    the minimum number of light-trails as the final solution $Sol$.
```

Fig. 4. LTA algorithm.

$$\binom{L_{\max}+1}{2} = \sum_{k=1}^{L_{\max}} k = \frac{1}{2}\left(L_{\max}^2 + L_{\max}\right), \qquad (5)$$

In *pack*(l), t_{ab} is first packed onto l. Then adjacent requests are considered in the descending order of h_{ij}. In case of multiple choices, d_{ij}^r and t_{ij} are compared in turn, and the requests with the largest d_{ij}^r or t_{ij} are packed first. Inside *pack*(l), we give h_{ij} higher priority than d_{ij}^r because d_{ij}^r has already been used to identify t_{ab} before entering *pack*(l).

The pseudo-code for *pack*(l) is given in Fig. 3. It returns a tentative packing scheme $\boldsymbol{P(l)}$, and the values of sum_h and sum_t as defined in (3) & (4).

D. LTA Algorithm

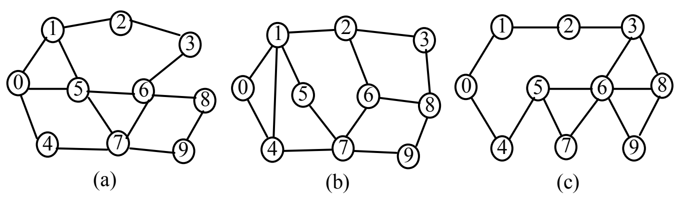
In LTA, we first use Floyd-Warshall algorithm [12] to calculate the all-pairs shortest distance matrix $\boldsymbol{H}=\{h_{ij}\}$. The *traffic matrix preprocessing* in Fig. 1 is used to ensure that all t_{ij} satisfies $h_{ij} \le L_{\max}$. Then, N solutions $\{\boldsymbol{Sol_r} \mid 0 \le r < N\}$ are constructed based on each r, where the most crucial request is considered first. Light-trails are determined from the packing process and formula (2). The final solution $\boldsymbol{Sol}$ is chosen from $\{\boldsymbol{Sol_r} \mid 0 \le r < N\}$ to minimize the number of light-trails required. LTA algorithm is formally summarized in Fig. 4.

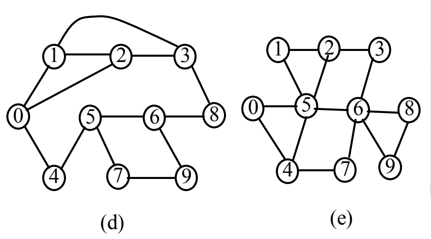
Assume that $\boldsymbol{L_T}$ is already obtained. We consider the time complexity of *Step 2)* in LTA. In constructing a light-trail (*Step 2b)~2e)*), at most $|\boldsymbol{L_T}|$ candidate trails are examined using the packing process, which considers $O(L_{\max}^2)$ requests (see Fig. 3 and formula (5)) and $O(L_{\max}^2)$ comparisons to determine a request for packing. Therefore, its complexity is $O(|\boldsymbol{L_T}|L_{\max}^4)$. Since each light-trail carries at least one request and there are at most N^2 requests, the complexity for an $\boldsymbol{Sol_r}$ construction is $O(|\boldsymbol{L_T}|L_{\max}^4 N^2)$. The final solution $\boldsymbol{Sol}$ is chosen among $\{\boldsymbol{Sol_r} \mid 0 \le r < N\}$. This gives *Step 2)* the complexity of $O(|\boldsymbol{L_T}|L_{\max}^4 N^3)$.

IV. Numerical Results

To compare the performance of LTA with ILP, we consider the examples shown in Fig. 5, where topology (a) and traffic matrix (g) are taken from [6], and traffic matrix (h) is obtained by enlarging the matrix in (g). Particularly, traffic matrix (g) is for topologies (a)~(e) and (h) is for topology (f). We assume $L_{\max}=4$ and $C=48$ for all examples.

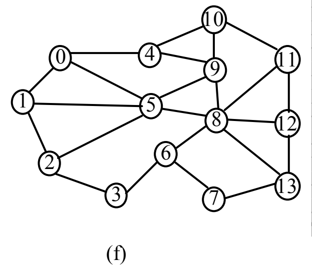
TABLE I summarizes the results. The 2nd and 3rd rows give the number of light-trails required by ILP and LTA. The 4th row shows the best r for LTA. Performance gap between LTA and ILP is given in the 5th row. The last two rows include the number of wavelengths/light-trails required at the most-heavily-loaded link. From TABLE I, we can see that LTA





$$\begin{bmatrix} 0 & 5 & 11 & 10 & 4 & 5 & 4 & 6 & 6 & 10 \\ 8 & 0 & 5 & 5 & 1 & 3 & 1 & 11 & 7 & 2 \\ 3 & 0 & 0 & 3 & 0 & 2 & 9 & 4 & 10 & 9 \\ 8 & 2 & 11 & 0 & 11 & 6 & 11 & 6 & 9 & 4 \\ 11 & 7 & 7 & 6 & 0 & 11 & 3 & 2 & 9 & 9 \\ 8 & 9 & 7 & 5 & 4 & 0 & 11 & 8 & 10 & 9 \\ 9 & 7 & 11 & 9 & 1 & 10 & 0 & 4 & 11 & 2 \\ 6 & 0 & 10 & 4 & 2 & 4 & 4 & 0 & 2 & 9 \\ 2 & 9 & 10 & 2 & 6 & 9 & 9 & 8 & 0 & 9 \\ 11 & 0 & 10 & 0 & 8 & 10 & 8 & 11 & 4 & 0 \end{bmatrix}$$

(g)



$$\begin{bmatrix} 0 & 5 & 11 & 10 & 4 & 5 & 4 & 6 & 6 & 10 & 2 & 5 & 1 & 1 \\ 8 & 0 & 5 & 5 & 1 & 3 & 1 & 11 & 7 & 2 & 3 & 2 & 1 & 2 \\ 3 & 0 & 0 & 3 & 0 & 2 & 9 & 4 & 10 & 9 & 8 & 3 & 1 & 1 \\ 8 & 2 & 11 & 0 & 11 & 6 & 11 & 6 & 9 & 4 & 9 & 6 & 3 & 2 \\ 11 & 7 & 7 & 6 & 0 & 11 & 3 & 2 & 9 & 9 & 1 & 10 & 1 & 12 \\ 8 & 9 & 7 & 5 & 4 & 0 & 11 & 8 & 10 & 9 & 11 & 1 & 12 & 3 \\ 9 & 7 & 11 & 9 & 1 & 10 & 0 & 4 & 11 & 2 & 6 & 16 & 6 & 1 \\ 6 & 0 & 10 & 4 & 2 & 4 & 4 & 0 & 2 & 9 & 2 & 4 & 8 & 8 \\ 2 & 9 & 10 & 2 & 6 & 9 & 9 & 8 & 0 & 9 & 8 & 1 & 10 & 1 \\ 11 & 0 & 10 & 0 & 8 & 10 & 8 & 11 & 4 & 0 & 2 & 1 & 1 & 7 \\ 8 & 6 & 4 & 5 & 18 & 11 & 7 & 1 & 8 & 1 & 0 & 2 & 1 & 3 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 0 & 13 & 3 \\ 2 & 6 & 1 & 5 & 1 & 1 & 5 & 3 & 9 & 1 & 10 & 2 & 0 & 8 \\ 1 & 11 & 2 & 9 & 8 & 1 & 17 & 1 & 2 & 1 & 1 & 2 & 1 & 0 \end{bmatrix}$$

(h)

Fig. 5. Numerical examples: topologies and traffic matrices.

TABLE I COMPARE LTA WITH ILP

Topology	(a) N=10	(b) N=10	(c) N=10	(d) N=10	(e) N=10	(f) N=14
ILP	13	14	15	14	14	27
LTA	14	15	16	15	16	30
Best r	2, 5, 7	3, 7	5	6	3	9, 13
Gap	7.69%	7.14%	6.67%	7.14%	14.29%	11.11%
W_{ILP}	4	3	4	5	4	4
W_{LTA}	4	4	5	5	5	5

TABLE II LTA SOLUTION FOR TOPOLOGY (a)

No.	Light-trail	Accommodated node pairs	Capacity
1	0→4→7→9→8	(4,9) (0,9) (4,8) (0,8) (7,8) (0,7) (9,8) (4,7)	48
2	8→9→7→4→0	(9,4) (9,0) (8,4) (8,0) (8,7) (7,0) (7,4)	43
3	9→7→6→3→2	(9,7) (9,2) (7,2) (9,6) (7,3) (7,6)	47
4	2→3→6→8→9	(8,9) (2,9) (3,9) (2,8) (6,9) (3,8) (2,3)	46
5	2→1→5→7→9	(7,9) (1,9) (2,7) (5,9) (1,7) (2,5) (5,7) (1,5)	48
6	9→8→6→5→1	(9,5) (8,1) (8,5) (6,1) (8,6)	44
7	4→0→5→6→3	(4,5) (4,3) (0,3) (4,6) (0,6) (5,3) (0,5)	44
8	1→0→5→6→8	(5,8) (1,8) (1,6) (6,8) (5,6) (1,0)	48
9	3→6→5→0→4	(5,4) (3,4) (3,0) (6,4) (6,0) (3,5) (0,4)	43
10	4→0→5→1→2	(4,0) (4,2) (4,1) (0,2) (5,2) (0,1)	48
11	3→6→7→5→1	(7,5) (3,7) (3,1) (6,7) (6,5) (3,6) (5,1)	46
12	8→6→3→2→1	(8,3) (8,2) (6,2) (6,3) (3,2)	43
13	2→1→5→0→4	(1,4) (2,0) (5,0)	12
14	0→1→2→3→6	(2,6) (1,3) (1,2)	19

returns a sub-optimal solution. Specifically, the LTA solution for topology (a) (with traffic matrix (g)) is detailed in TABLE II. Compared to the ILP solution in [6], the number of light-trails required by LTA is only one more than ILP.

In order to evaluate the efficiency of the reference node mechanism, we compare LTA with an algorithm LTD without reference node. LTD is obtained by slightly modifying LTA as follows: 1) *Step 2a)* in Fig. 4 is removed; 2) In *Step 2b)*, the most crucial request t_{ab} is identified as the one with the largest h_{ij} and (then) t_{ij}, and d_{ij}^{r} is removed; 3) Both *Step 2g)* and *Step 3)* are removed and LTD ends at *Step 2f)*, because only one solution is constructed; and 4) d_{ij}^{r} is removed from the packing process in Fig. 3.

TABLE III LTA VS LTD BASED ON 100 RANDOM TESTS ($L_{\max}$=4, C=48)

N	10	15	20	25	30	35	40
LTA_{best}	12.85	32.90	66.25	114.94	183.79	270.73	380.57
LTA_{ave}	13.31	34.34	68.97	119.41	189.66	278.94	390.50
LTD	13.82	34.72	69.12	119.30	189.15	278.06	388.91

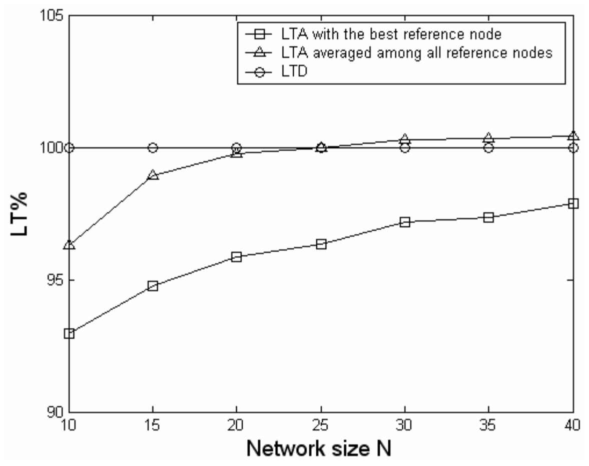


Fig. 6. Reference node efficiency.

TABLE III and Fig. 6 give the results averaged over 100 randomly generated topologies and traffic matrices. The topologies and the traffic matrices are generated as follows.

- *Topology construction*. We first treat the topology as a directed graph. Each node emanates 2 directed edges to others with the same probability. Then, each directed edge is replaced by a bidirectional edge. We also check the topology to ensure it is connected.
- *Traffic matrix generation*. $\boldsymbol{T}=\{t_{ij}\}$ is first generated with t_{ij} uniformly distributed in [0, 12]. Then we adjust $\boldsymbol{T}$ to characterize some "hot" paths as follows. If t_{ij}=12, we reset it to 0; if t_{ij}=11, we replace it by another random number in [0, 30]. Then, the traffic matrix preprocessing in Fig. 1 is applied to $\boldsymbol{T}$. For simplicity, we set t_{ij}=0 if it is greater than C=48 after traffic matrix preprocessing.

In TABLE III, the 2nd and the 3rd rows give the number of light-trails required by LTA *using the best reference node* and *averaged over the N solutions* $\{\boldsymbol{Sol_r} \mid 0 \le r < N\}$, respectively. LTD results are given in the 4th row. TABLE III also translates into Fig. 6, where percentage is used with 100% for LTD. From Fig. 6, we can see that the gain of LTA over LTD is more significant for small networks. For N>25, though LTA_{ave} is a little bit larger than LTD, LTA_{best} is still much smaller. This is because request discreteness in small networks can be efficiently suppressed by reference node. For large-sized networks, the efficiency fades out.

V. DISCUSSION

Generally, it is difficult to determine the best reference node r. Fig. 7 gives an example with $L_{\max}$=3 and C=48. We first assume r=0. Note that d_{ij}^{r}=3 and h_{ij}=1 for all the requests. In Fig. 7a, two light-trails are required to carry $\{t_{12}$=30, t_{34}=18$\}$

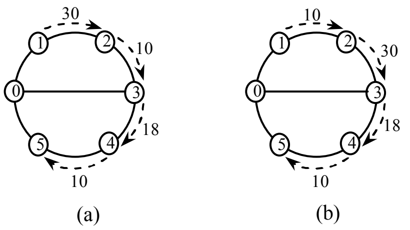


Fig. 7. The best reference node depends on both topology and traffic matrix.

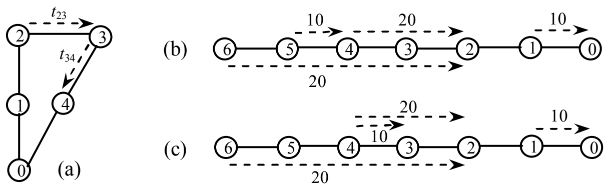


Fig. 8. Intelligence of d_{ij}^r definition in (1) (L_{max}=4, C=48, r=0).

and {t_{23}=10, t_{45}=10} according to LTA. But, if we exchange t_{12} and t_{23} as in Fig. 7b, then a light-trail is required to carry {t_{23}=30, t_{34}=18}, and {t_{12}=10} and {t_{45}=10} need two additional light-trails due to request discreteness. On the other hand, if r=5, only two light-trails are required for both cases in Figs. 7a & 7b. This example shows that the best reference node depends on both topology and traffic matrix.

Two-dimension coordinates can do a better job in quantifying request discreteness. For example, the discreteness among the requests with the same d_{ij}^r is ignored in Fig. 7. Therefore, the reference node mechanism in LTA can only *partially* address the request discreteness issue. On the other hand, introducing two-dimension coordinates will dramatically complicate the algorithm.

We define $d_{ij}^r=h_{ir}+h_{jr}$ as in (1) instead of $d_{ij}^r=\max\{h_{ir}, h_{jr}\}$. With the definition $d_{ij}^r=h_{ir}+h_{jr}$, we can identify the most crucial request in a more intelligent way than $d_{ij}^r=\max\{h_{ir}, h_{jr}\}$. This is shown by the three scenarios in Fig. 8, where L_{max}=4, C=48 and r=0. In Fig. 8a, t_{23} *physically* locates farther away from node 0 than t_{34}. In this case, $d_{ij}^r=h_{ir}+h_{jr}$ can properly return t_{23} as the most crucial request. But with $d_{ij}^r=\max\{h_{ir}, h_{jr}\}$, this will fail because d^0_{23} and d^0_{34} will take the same value of 2.

Figs. 8b & 8c further show the intelligence of our d_{ij}^r definition. The two scenarios differ from each other only by the location of one request: t_{54} in Fig. 8b and t_{43} in Fig. 8c. With $d_{ij}^r=h_{ir}+h_{jr}$, t_{54} (instead of t_{62}) in Fig. 8b is identified as the most crucial request, whereas t_{62} is identified in Fig. 8c. Since t_{54} is the most crucial request in Fig. 8b, it shares a common light-trail with t_{62}. As a result, the remaining not-yet-carried requests t_{42} and t_{10} can share a common light-trail. Similarly, since t_{62} is the most crucial request in Fig. 8c, {t_{62}, t_{42}} and {t_{43}, t_{10}} require two light-trails. In both scenarios, two light-trails are sufficient. However, if d_{ij}^r is defined by $d_{ij}^r=\max\{h_{ir}, h_{jr}\}$, then t_{62} will be the most crucial request in both Figs. 8b & 8c. According to the packing process *pack*(l) in Fig. 3, t_{62} in Fig. 8b will share a light-trail with t_{42} instead of t_{54}. As a result, the remaining not-yet-carried requests t_{54} and t_{10} need two additional light-trails (instead of only one). Compared with $d_{ij}^r=h_{ir}+h_{jr}$, the definition $d_{ij}^r=\max\{h_{ir}, h_{jr}\}$ leads to a worse solution for Fig. 8b.

The reason behind the above differences is as follows. A request with smaller h_{ij} (such as t_{54} in Fig. 8b and t_{43} in Fig. 8c) generally has less opportunity to be carried by the current light-trail in the packing process *pack*(l). If it is far from r (such as t_{54} in Fig. 8b), then it may boost request discreteness. Therefore, we should give it higher priority and accommodate it first. Although t_{62} *physically* locates farther away from r=0 than t_{54}, it is advantageous to have a larger h_{ij} and thus being preferred by *pack*(l).

In LTA, we have constructed all the light-trails with the same length limit L_{max}. We can check through the final solution to see if some light-trails can be shortened. For example, in Fig. 7b, the light-trail carrying {t_{23}=30, t_{34}=18} can be shortened to 2 hops although L_{max}=3.

VI. Conclusion

In this paper, we analyzed the key factors that affect the solution of the light-trail assignment problem. These factors include the request discreteness, the shortest path length of each request, and the flow volume. Taking all the three factors into account, we proposed an intelligent heuristic algorithm LTA (Light-Trail Assignment) to minimize the number of light-trails required. Notably, LTA adopts a reference node mechanism to address the request discreteness issue. To the best of our knowledge, LTA is the first heuristic proposed in this area. It was shown to be very efficient by always returning a sub-optimal solution.

References

[1] M. Mahony, D. Simeonidou, D. Hunter and A. Tzanakaki, "The application of optical packet switching in future communication networks", *IEEE Communications Magazine*, pp. 128-135, Mar. 2001.

[2] C. Qiao and M. Yoo, "Optical burst switching (OBS) – a new paradigm for an optical internet", *J. High Speed Networks (JHSN)*, vol. 8, issue 1, pp. 69-84, 1999.

[3] A. Gumaste and I. Chlamtac, "Light-trails: a novel conceptual framework for conducting optical communications", *IEEE HPSR '03*, pp. 251-256, Jun. 2003.

[4] I. Chlamtac and A. Gumaste, "Light-trails: A solution to IP centric communication in the optical domain", Online publication: February 17, 2003, Springer-Verlag Berlin Heidelberg 2003.

[5] M. T. Frederick, N. A. VanderHorn, and A. K. Somani, "Light trails: a sub-wavelength solution for optical networking", *IEEE HPSR '04*, pp. 175-179, Apr. 2004.

[6] Jing Fang, Wensheng He, and A. K. Somani, "Optimal light trail design in WDM optical networks", *IEEE ICC '04*, vol. 3, pp. 1699-1703, Jun. 2004.

[7] A. Gumaste, G. Kuper and I. Chlamtac, "Optimizing light-trail assignment to WDM networks for dynamic IP centric traffic", *IEEE LANMAN '04*, pp. 113-118, Apr. 2004.

[8] A. Gumaste, I. Chlamtac, and J. P. Jue, "Light-frames: a pragmatic framework for optical packet transport", *IEEE ICC '04,* vol. 3, pp. 1537-1542, Jun. 2004.

[9] Yabin Ye, H. Woesner, R. Grasso, Tao Chen and I. Chlamtac, "Traffic grooming in light trail networks", *IEEE GLOBECOM '05*, vol. 4, pp. 1957-1962, Dec. 2005.

[10] Weiyi Zhang, Guoliang Xue, Jian Tang and K. Thulasiraman, "Dynamic light trail routing and protection issues in WDM optical networks", *IEEE GLOBECOM '05*, vol. 4, pp. 1963-1967, Dec. 2005.

[11] D. Kliazovich, F. Granelli, H. Woesner and I. Chlamtac, "Bidirectional light-trails for synchronous communications in WDM networks", *IEEE GLOBECOM '05*, vol. 4, pp. 1947-1951, Dec. 2005.

[12] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Upper Saddle River, NJ: Prentice-Hall, 1993.