All-Optical Multicast Routing Algorithms without Splitters

Dinh Danh Le, Miklós Molnár, Jérôme Palaysi

All-Optical Multicast Routing Algorithms without Splitters

Dinh Danh Le, Miklós Molnár, and Jérôme Palaysi

University Montpellier 2, LIRMM
161 rue Ada, 34095 Montpellier Cedex 5, France
{Dinhdanh.Le,Miklos.Molnar,Jerome.Palaysi}@lirmm.fr
http://www.lirmm.fr

Abstract. Multicasting in WDM core networks is an efficient way to economize network resources for several multimedia applications. Due to their complexity and cost, multicast capable switches are rare in the proposed architectures. In practical routing cases, the state of the network is given by a directed graph. The paper investigates the multicast routing without splitters in directed asymmetric topologies. The objective is to minimize the number of used wavelengths and if there are several solutions, choose the best cost one. We show that the optimal solution is a set of light-trails. The problem is NP-hard even in symmetric digraphs. An efficient heuristic is proposed to minimize the conflict between the light-trails, and so to minimize the number of used wavelengths. The performance is compared to existing light-trail based heuristics and our algorithm provides a good solution with a few wavelengths required and a low cost.
Keywords: WDM network, multicast routing, multicast incapable node, light-trail, wavelength minimization, heuristic

1 Introduction

All optical networks are serious candidates to become high speed backbone networks with huge capacity. In optical routing, the messages are transmitted by light signal without electronic processing. Routes should satisfy the physical (optical) constraints in static connection based networks and also in the case of burst and packet switching.

Multicast communications presents in networks to efficiently perform data transmission from a source to several destinations. Usually, multicast routes corresponds to trees in the topology graph (there can be light-trees in WDM networks). To realize, there must be multicast capable nodes (splitters) at all the branching nodes of the tree. However, one of the most hard constraints for optical multicasting is the constraint on the availability of light splitters in the switches. Splitters are expensive and the light power can be decreased considerably by splitting (inversely proportional with the number of outgoing ports [2]).

In our paper, we investigate on the interesting question: how to perform multicast without splitters. Trivially, a set of light-paths from the source to the destinations can be used as a solution, but this solution is expensive in term of light-paths. Our objective is to perform the multicasting without splitters and minimizing the number of used wavelengths. Solutions in symmetric networks (when one can suppose the availability of a wavelength in both directions between the connected switches) are known, but we investigate the asymmetric case when the network topology is an asymmetric directed graph (or mixed graph). This case is very practical in reality. Even if the network is designed to be undirected, when some demands arrive and hold some of the resources of the network, the resulting network graph is now mixed, therefore the routing for subsequent demands will be calculated on a mixed-graph network. Another case where mixed-graph routing is required is when two arc-disjoint trees must be found on a graph (e.g., for protection against a single-link failure scenario). If the calculation of the primary and secondary trees is done sequentially, the secondary tree, after the removal of the primary one, will be calculated on a mixed graph. The multicast routing in this case can also be well applicable in the dynamic routing when multiple requests are coming and leaving with arbitrary period of time.

Some studies indicated that non simple light-trail (corresponding to non simple walks) can be used for multicasting [1] if the TaC option is employed in the cross-connects (OXCs) and crossing an OXC several times by the same wavelength is possible. In the paper, we show that the optimal route minimizing the number of wavelength is a set of (non simple) light-trails. The computation of the optimum in asymmetric graphs is hard. So, we propose some heuristic algorithms, which try to minimize the number of wavelengths, taking into account the availability of wavelengths, with a reasonable cost. We compare the performance of the algorithm with two previously proposed light-trail based multicast routing algorithms.

The structure of the paper is the following. Section 2 presents the considered problem and some related ones. The most important related works are mentioned in Section 3. Some used concepts and properties are given in Section 4. Our heuristic is described in Section 5 followed by the experimental results in Section 6. We summary our work and discuss about the future works in the Section 7.

2 Problem Formulation

The considered network is modeled by the topology graph $G = (V, A)$ is an arbitrary directed graph (or digraph) in which each arc represents the availability of a fiber between the pair of nodes and there are at most two fibers between any pair of nodes. This configuration is realistic in real networks. We suppose that each fiber has the same set of available wavelengths and each arc $e \in A$ is associated with the a positive value $cost(e)$. Given the multicast request $r = (s, D)$, in which $s \in V$ is the source node and $D \subseteq V \setminus \{s\}$ is the set of destinations, the routing problem is to compute the routes to perform multicast for r.

In this study, we work on the networks in which the nodes are not equipped with any splitters but TaC-cross connects that allow signal to tap the local station with a small power and forward the remaining to one of the output ports. Besides, the nodes can be traversed by the same wavelength several times as long as there are different incoming and outgoing ports for each pass. So, not light-trees but light-trails from the source to the destinations can perform the multicast. To ensure a feasible solution, we suppose that there is at least one directed path from the source to each destination.

Let T be the set of computed light-trails $t_i, i = 1, \ldots, k\}$ for the request r, we define the total cost as the summation of all the cost of them, given by: $TotalCost(T) = \sum_{i \in [1,k]} \sum_{e \in t_i} cost(e)$.

To perform the routing respecting the *distinct wavelength constraint*[1], each fiber is assigned several wavelengths such that the number of assigned wavelengths is equal to the number of conflict trails passing it. The number of wavelengths needed to perform the routing is equal to the maximum number of wavelengths that are assigned for one fiber.

Different objectives for the multicast routing can be formulated as follows.

Problem 1 (Routing using a minimum number of wavelengths).
Instance: a network G, a source node s and a set of the destination nodes D
Solution: a set of light-trails T rooted at s and covering all the destinations
Objective: minimize the number of wavelengths used by T

Problem 2 (Minimum cost routing).
Instance: a network G, a source node s and a set of the destination nodes D
Solution: a set of light-trails T rooted at s and covering all the destinations
Objective: minimize the total cost of T

In both cases (with the given constraints), the optimal solution is a set of light-trails routed at the source and covering all of the destinations. Notice that this set corresponds to a hierarchy obtained from a star (cf. [6] for the definition of a hierarchy). The solution of Problem 1 can be composed from very long trails. The optimum of Problem 2 can use a high number of wavelengths. Trade-off can be interesting.

Problem 3 (Minimum cost multicast routing using a given number of wavelengths).
Instance: a network G, a source node s and a set of the destination nodes D, the number of wavelengths $W \in \mathbb{Z}^+$
Solution:a set of light-trails T rooted at s and covering all the destinations and using at most W wavelengths
Objective: minimize the total cost of T

Problem 4 (Length limited multicast route using a minimum number of wavelengths).

[1] *Distinct wavelength constraint:* Different light-paths or light-trees sharing the common link must be allocated distinct wavelengths [11].

Instance: a network G, a source node s and a set of the destination nodes D, the number $L \in \mathbb{Z}^+$
Solution: a set of light-trails T rooted at s and covering all the destinations, and its length (the number of hop counts from s to the terminal d_i) does not excess L
Objective: minimize the number of wavelengths used by T

The solutions of the last two problems are also sets of light-trails.

The mentioned routing problems are hard optimization problems. Problem 1 corresponds to finding the solution with minimal number of colors to assign the trails such that two trails shared a common arc must be assigned with two different colors. Problem 2 is equivalent the Degree Constrained Directed Minimum Spanning Tree Problem in the distance graph of the problem.

In our study, we focus on the Problem 1, but try to find a solution with the lowest total cost. That is, we first try to minimize the number of used wavelengths, then try to minimize the cost among the solutions with the same minimal wavelengths. Some results can be useful to solve Problem 3 and Problem 4.

2.1 Hardness of the problem

In this subsection we prove that the Problem 1 (P_1) given above is NP-hard. We first consider the following problem and prove that it is NP-complete.

Problem (One Spanning Trail - OST)
Instance: A directed graph $G' = (V, A)$ and a pair (s, D)
Question: Is there a spanning trail from s covering all the destinations in D?

We start with the problem of two arc-disjoint paths stated as follows:

Problem(Two Arc-Disjoint Paths - 2ADP)
Instance: A directed graph G and two pairs of vertices (x, x') and (y, y')
Question: Does exist a path between x and x' and an other path between y and y' that are arc-disjoint?

Since problem 2ADP is a particular case of the NP-complete Subgraph Homomorphism Problem with the pattern of two disjoint paths [4], so it is NP-complete. In the following, we prove the NP-hardness of the Problem OST by transforming from the Problem 2ADP.

Let G and two pairs of vertices (x, x') and (y, y') be any given instance I of Problem 2ADP, we create a graph G' and adding:

- three vertices: s, d', d'';
- the arcs $(s, x), (x', d'), (d', y)$ and (y', d'')
- and we set $D = \{d', d''\}$

So G' and a pair (s, D) form an instance I' of Problem OST.

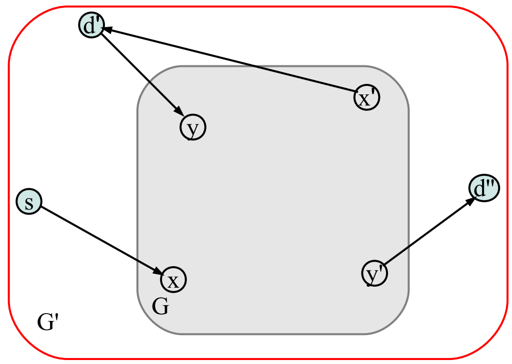


Fig. 1. Illustration of the two problems OST and 2ADP

It is easy to verify that the transformation is polynomial. We now prove that there exists a solution S for the instance I of Problem 2ADP if and only if there exists a solution S' for the instance I' of Problem OST.

1. Suppose that two arc-disjoint paths $P = (x_0, x_1, ..., x_k), Q = (y_0, y_1, .., y_l)$ in which $x_0 = x, x_k = x', y_0 = y, y_l = y'$, be a solution S for Problem 2ADP. Let S' be a trail of the form $(s, x, x_0, x_1, ..., x_k, d', y_0, y_1, .., y_l, d'')$ (Figure 1). Obviously, S' needs only one color, so S' is a solution for the Problem OST.
2. In contrast, suppose that there exists a solution (a trail) S' for Problem OST. S' is of the form: $(s, x, x_0, x_1, ..., x_k, d', y_0, y_1, .., y_l, d'')$, in which $P = (x_0, x_1, ..., x_k), Q = (y_0, y_1, .., y_l)$, $x_0 = x, x_k = x', y_0 = y, y_l = y'$. Because S' needs just one color, all the arcs occur once, so the two paths P, Q are arc-disjoint. Thus, P, Q form the solution S for the Problem 2ADP.

Because 2ADP is NP-complete, OST is NP-complete. The Problem OST is the special case of the more general Problem P_1 - What is the minimum number of colors? - so Problem 1 is NP-hard, but not NP-complete because it is not a decision problem.

3 Related Work

Due to its interest, WDM multicast routing has been investigated intensively in the literature and several propositions exist to adapt multicast routing algorithms to the optical constraints (cf. [9] for some basic algorithms and [11] for a survey). The minimization of the number of used wavelengths was investigated at first in [5] in which the wavelengths are supposed to unevenly distribute in the networks. The considered network is assumed to be equipped with splitters and wavelength converters. The multicast is based on a tree. The objective is to construct a tree T meeting optical constraints such that the number of wavelengths used to cover T is minimized. The NP-hardness of the problem is proved and an approximation algorithm has been proposed. An improved approximation can be found in [8].

The case of switching without splitters in symmetric networks has been discussed in [1]. The problem is to find a Multiple-Destination Minimum Cost Trail (MDMCT) that starts from a source and spans all the destinations with minimizing the summation of costs of directed edges traversed (the total cost). To ensure a feasible solution, a low-cost cross-connect architecture called Tap-and-Continue (TaC) has been proposed to replace splitters. TaC cross-connects can tap a signal with small power at the local station and forward it to one of its output ports. Moreover, every link is assumed to have at least two fibers in order to support bidirectional transmission on the same link. The solution is a route corresponding to a light-trail that can be non simple.

At first, the authors proved that the MDMCT problem is NP-hard and then developed a heuristic (called MDT) that finds a feasible trail in polynomial time. The algorithm has two steps. The first step is finding an approximated Steiner tree for a multicast request using the Minimum Cost Path Heuristic (MCPH) proposed in [7]. A trail then is computed based on the backtracking method following the tree.

The advantage of MDT heuristic is that it uses only one wavelength (and one transmitter) for each multicast request (and thus, the wavelength is minimized). However, because of multitude of round-trip traversing, a large number of links in both directions is required, hence the total cost and the diameter of the light-trail can be very high. To improve the total cost, it is necessary to reduce the round-trip traversing. Moreover, it is worth noting that, the source can inject the light signal by multiple transmitters independently. By taking this feature into account, one can considerably reduce the reversal arcs (that reroutes to the source), then the total cost and the diameter can also be reduced. This is the idea to make a modified version of MDT, called MMDT that is detailed in Section 5.

In [3], Der-Rong Din also posed the problem of finding the routing tree(s) which minimizes the cost under WDM symmetric networks using only TaC cross-connects. This problem is named Minimal Cost Routing Problem (MCRP). Unlike the approach of [1] that based on light-trail, the approach of Der-Rong Din is based on light-forest with multiple transmitters are implicitly employed. Furthermore, to produce a trade-off between the total cost of the light-forest and the number of wavelength, the cost is defined as a function which combines two metrics: the actual total cost of the light-forest and the cost for using wavelengths, in which second component is calculated by a coefficient α times the number of used wavelengths (cf. [3] for more details).

The author proposed two heuristic algorithms, namely Farthest-Greedy (FG) and Nearest-Greedy (NG). The two algorithms are based on the shortest path tree (SPT). The idea of these algorithms is: first construct the SPT from the source to the destinations, then keep one path for each subtree of the source, and finally reroute the other destinations that have not been reached (*unreached* destinations). The difference between the two algorithms is: FG keeps the farthest (maximal cost) destination routed by the computed shortest path, and choose the farthest destination in the unreached set to reroute, whereas NDF keeps the nearest (minimal cost) destination and choose the nearest destination in unreached set to reroute in the rerouting phase. The rerouting phase of FG and NG is performed by the shortest paths from the source or from the leaves of computed trees to each unreached destination, that do not share any nodes and edges with all the computed trees (each tree is computed in a different wavelength graph that is initialized by the original graph). When there is no possible path in the computed trees or the path exists but with larger cost than the path found in the new wavelength graph, the unreached destination is routed by the shortest path found in the new tree with a new wavelength. The author also gave the comparison between FG, NG and MDT by simulations, and the results show that FG outperforms NG and MDT.

Almost the solutions proposed in the literature (excluding MDT) are based on simple routes in which cycles are not allowed. However, one can operate multicasting by non simple routes which permit nodes to be visited several times, as long as the routes using the same wavelength are arc-disjoint. MDT in [1] gives a special structure which allow cycles, but with special cycles which are 2-cycles[2]. In fact, one can construct structures that allow not only 2-cycles but also arbitrary cycles. These structures correspond to a hierarchy that was proposed in [6]. For multicast routing in WDM networks, the light-hierarchy concept has been proposed in [10]. A light-hierarchy is a hierarchy using a single wavelength. In this study, we create light-trails to multicast without the need of splitters.

4 Useful definitions and properties

In order to describe our algorithm, some concepts should be given in the following.

Directed spanning tree (DST): A directed tree rooted at the multicast source covering all the destinations. In our algorithm, we use two kinds of DST: Directed shortest path tree (DSPT) and Directed Approximation Steiner tree (DAST). A DSPT is a directed tree composed by the shortest paths from the source to the destinations. To compute DSPT, any shortest path algorithms can be employed (e.g., Dijkstra algorithm) ensuring that the shortest paths are loop-free. In contrast, a DAST can be computed by employing the Minimum Path Heuristic proposed in [7]. In our algorithm, one of these DSTs is computed in the first step, from it the light-trails are constructed.

Conflict graph: A graph used to represent the conflicts among the trails. Formally, in our study, a conflict graph is $G_C = (T, E)$, in which T is a set of nodes corresponding to the trails and E is a set of edges such that $e = \{t, p\} \in E$ if and only if there is a conflict between trail t and trail p, i.e.,

[2] In graph theory, an n-cycle is a cycle with n vertices.

two trails share a common arc. In this study, we just consider conflicts such that shared arcs form the prefix of the concerned trails and this property is preserved during the algorithm. For this condition, each of connected components in the conflict graph corresponds to a subtree of the DSPT (Figure 2).

Property 1: Each connected component in the conflict graph composes a (conflict) clique[3].
Trivially, if the shared arcs of conflicting trails are the prefix of the trails, then conflicts are transitive (if there is a conflict between T_1 and T_2 and between T_2 and T_3, then there is a conflict between T_1 and T_3). Indeed, all the trails in the same connected component share the first arc from the source. So the Property 1 follows.

Property 2: The number of colors needed to color all the nodes of a clique is equal to the number of nodes of that clique.
Obviously, there is a (conflict) edge between every pair of nodes in each clique. To avoid the conflict, the nodes must be colored with different colors. So the Property 2 follows.

Property 3: The number of wavelengths needed to perform the routing respecting the *distinct wavelength constraint* in network fibers is equal to the number of nodes (the size) of the maximal clique in the conflict graph in our study case.
In deed, each clique corresponds to a subtree of the DSPT. These subtrees are arc-disjoint, so the corresponding trails in each clique do not share any arcs with the corresponding trails in the other ones. Thus, the minimal number of colors needed to color all the nodes in the conflict graph is equal to the size of the maximal clique, because we can use some colors that have been used in the maximal clique to re-color the other nodes in the other cliques. Moreover, to guarantee the distinct wavelength constraint, the number of colors needed in each clique is equal to the number of wavelengths needed to assign the corresponding trails in that clique. So this property holds.

Thus, the problem of minimizing the number of used wavelengths reduces to the problem of minimizing the number of nodes (trails) of the maximal clique in the conflict graph.

Figure 2 illustrates a set of paths (trails) composing a DSPT for the multicast request $r = (s, \{d_1, d_2, d_3\})$ and the corresponding conflict graph. In Figure 2, there are two cliques corresponding to two subtrees of the DSPT. The maximal clique is composed from the paths T_1, T_2, T_3 starting from the source to the destinations d_1, d_2, d_3, respectively. It needs three wavelengths to color the three trails. The other clique composed from only the path T_4 that can re-use one wavelength that were assigned for the maximal clique.

[3] A clique of a graph is a complete subgraph of that graph

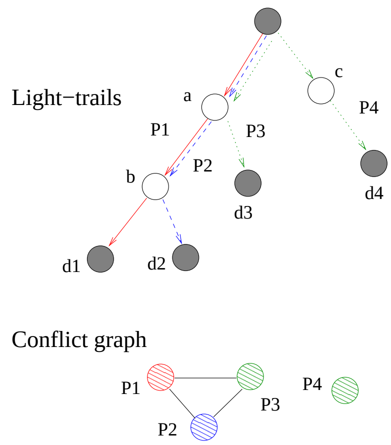


Fig. 2. Example of a DSPT and its conflict graph

5 Proposed Heuristics to Minimize the Wavelengths

5.1 The Algorithm Framework

The idea of the algorithm is to diminish the number of nodes (trails) in the maximal clique of the conflict graph until it cannot be reduced. Informally, the algorithm starts from a set of directed trails (at first, simple trails or paths in the DST). Then it tries to iteratively diminish the number of trails in the maximal clique, says C_{max}. At each step, it chooses one trail from the maximal clique that can be replaced by another trail, says T_0. Some mechanism can be employed to select this trail. When the trail was selected, the algorithm looks for all the other trails in the set of trails and choose the one (say, T_k) such that the terminal of it has the arc-disjoint shortest path to the first destination of the trail T_0. Then T_0 is replaced by the corresponding trail T_k, and the cardinality of the C_{max} is reduced by 1. The algorithm iteratively until the maximal clique cannot be reduced.

The framework of the algorithm consists of four main steps that can be described as follows:

The Algorithm Framework

Step 1: Compute a directed shortest path tree (DSPT) from the source s covering all destinations. If there is no branching node in the DSPT (except the source), then DONE. Otherwise, do Step 2.

Step 2: Compute the conflict graph from the DSPT. Each conflict clique corresponds to a sub-tree rooted at the source of the DSPT.

Step 3: Repeat Step (3.1) to Step (3.4) in the following until the cardinality of the maximal clique C_{max} cannot be reduced (is equal to 1 for the best case).

Step 3.1: Find the maximal clique C_{max}[4].

Step 3.2: Choose a trail T_0 in C_{max} with some mechanism that is mentioned below. Calculate the terminal l_0 of T_0 and the first destination f_0[5] of T_0. For example, in Figure 2 a), C_{max} comprises the paths T_1, T_2, T_3; the selected trail T_0 is T_3, l_0 is node d_3 and f_0 is also d_3.

Step 3.3: For every terminal l_i of the remaining trails T_i in the set of trails (except T_0), compute the trail T_k such that the path (l_k, f_0) is *arc-disjoint* with all the current trails, and it is the shortest path among the paths (l_i, f_0). If there is an arc-disjoint path from the source s to f_0, then take the shorter one between (l_k, f_0) and (s, f_0). In Figure 2 a), T_k is T_4, and l_k is node d_4.

Step 3.4: Graft the path (l_k, f_0) and the path (f_0, l_0) to the trail T_k, set l_0 as the terminal of the new trail T_k, remove the trail T_0, reduce the cardinality of clique C_{max} by 1. If the path (s, f_0) exists and is selected, create a new trail (s, l_0) and remove the trail T_0.

Step 4: Record the trails, the cardinality of clique C_{max} as the minimum number of wavelengths required. Employ the trail-wavelength-assignment (TWA) algorithm (described below) to assign wavelengths for the set of final trails.

Trail-wavelength-assignment (TWA) algorithm

The TWA algorithm mentioned in Step 4 works as follows. Let k be the minimum number of wavelengths returned by the routing algorithm above, and $w_1, w_2, .., w_k$ be the k wavelengths reserved for the multicast request. With each wavelength $w_i, i = 1, .., k$, assign w_i for every clique in the set of the remaining cliques, one trail for each clique. Repeat that until there is no trail in the set of remaining cliques.

Two greedy variants

In step 3.2, two greedy mechanisms for selecting the first trail T_0 in the maximal clique C_{max}, resulting in two variants of our algorithm, namely Farthest First (FF) if T_0 is the farthest trail (in term of cost, i.e. $cost(T_0) \rightarrow max$) among all the other trails in C_{max}; and Nearest First (NF) if T_0 is the nearest trail ($cost(T_0) \rightarrow min$). The descriptions of the two variants are shown in the following.

[4] To accelerate this step, C_i is organized in a priority queue in which the priority value is the size of C_i, and only cliques C_i with the size larger than 1 are pushed into the queue

[5] f_0 is the first destination on the path from the nearest branching node of l_0 to l_0

5.2 Farthest First

Farthest First

Input: A directed weighted graph G, number of available wavelengths W, a multicast request $r = (s, D)$.
Output: A set of light-trails T satisfying r
Objective: Minimize number of wavelengths.

Step 1:
construct a DSPT from s covering all destinations in D.
add every path from s to each leaf of DSPT to T. If there is no branching node in T, then DONE. Otherwise, do Step 2.
Step 2:
calculate cliques $C_i, i = 1, 2, .., n_c$, n_c is the number of sub-trees rooted at the child-nodes of s in the DSPT.
put C_i with $|C_i| > 1$ into a priority queue PQ ($PQ = \{(C_i, |C_i|)\}$)
Step 3:
$G' \leftarrow G \setminus T$ $\{G'$ is the remaining graph by removing arcs corresponding to all the trails in $T\}$
while $PQ \neq \emptyset$ **do**
  $\{$*Step 3.1:*$\}$
  pop the maximal clique C_{max} from PQ
  sort the trails in C_{max} descending of their costs
  $\{$*Step 3.2:*$\}$
  for each trail T_0 in the sorted C_{max} **do**
    let l_0 be the the terminal of T_0, calculate the first destination f_0
    find the shortest arc-disjoint path from the source s to f_0 in G' (if any).
    for every terminal l_i of trail T_i **do**
      find the shortest arc-disjoint path from l_i to f_0 in G'
    end for
    record the terminal l_k of the found trail.
    if either s or l_k is found **then** go to *Step 3.3*
  end for
  $\{$*Step 3.3:*$\}$
  if neither s nor l_k is found for all trails in C_{max} **then**
    go to Step 4 to finish
  else
    if $cost_{P(s,f_0)} \leq cost_{P(l_k,f_0)}$ **then** $T_j \leftarrow (s, l_0)$; $T \leftarrow T \cup \{T_j\}$
    else $T_k \leftarrow T_k \cup (l_k, f_0) \cup (f_0, l_0)$; $T \leftarrow T \cup \{T_k\}$
    $T \leftarrow T \setminus \{T_0\}$; $|C_{max}| = |C_{max}|$ - 1.
    if $|C_{max}| = 1$ **then** go to Step 4 to finish
    else update the new size $|C_{max}|$ for the C_{max} in PQ
  end if
end while
Step 4:
if $|C_{max}| > W$, **then return** FALSE

else record $|C_{max}|$ as the minimum number of wavelengths required.
employ the TWA algorithm to assign wavelengths for the set of final trails.

5.3 Nearest First

The description of NF is the same FF, except that in the *Step 3.1*, the trails in the C_{max} is sorted according to the ascending of their costs, in order to choose the feasible trail as near as possible.

If we use DAST instead of DSPT in Step 1 of FF, we have another heuristic call STFF. In the same way, if we use DAST for NF, we have STNF.

In summary, we have four heuristic variants of our algorithm: two are DSPT based (FF, NF) and other two are DAST based (STFF, STNF). The four heuristics produce a set of light-trails forming a special hierarchy (cf. [6]). Their performances are evaluated and compared with other algorithms in the next section.

5.4 Algorithm Illustration

In order to demonstrate the algorithm, we use a network in Figure 3 as an example. Moreover, due to the limited space, and because FF and NF have the same principle, we just illustrate the heuristic FF in the Figure 4 below.

After the Step 1 and Step 2, the DSPT and the initial conflict graph are shown in Figure 4 a). The maximal clique comprises three paths T_{10}, T_{12}, T_{13}, in which T_{12} is the farthest one, so it is selected first. The first destination f_0 of T_{12} is node 8, the shortest arc-disjoint path computed is the path passing the nodes {10, 5, 8}. Thus T_{12} is replaced by T_{10}, the new trail is $T_{12}^{'}$ (Figure 4 b)). Similarly, $T_{12}^{'}$ is then replaced by $T_{12}^{''}$ in the next run (Figure 4 c)). The final set of trails are shown in (Figure 4 d)).

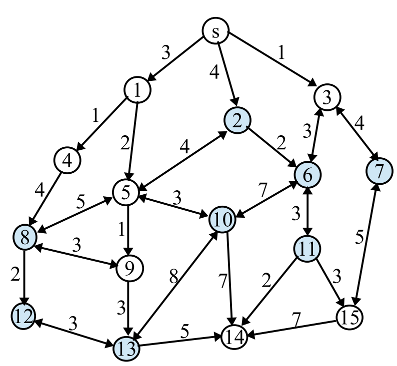
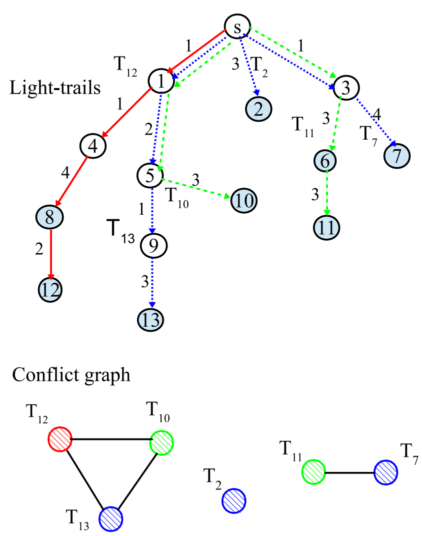
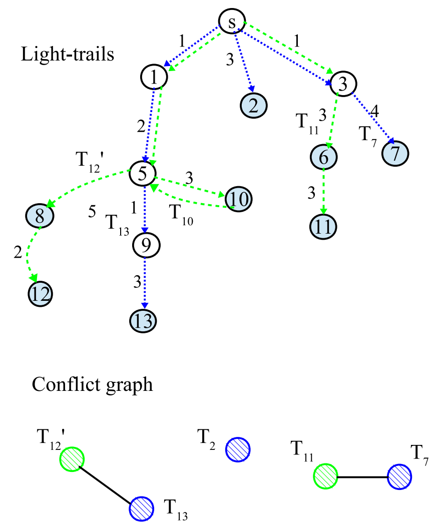


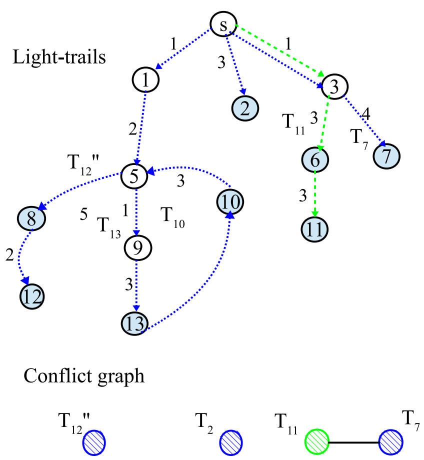
Fig. 3. A digraph to consider



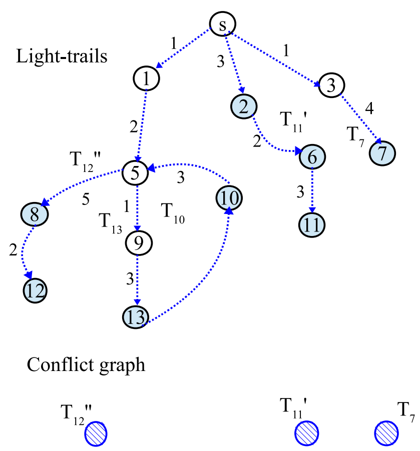
(a) The DSPT and the initial conflict graph



(b) Replace T_{12} by T'_{12}



(c) Replace T'_{12} by T''_{12}



(d) Replace T_{11} by T'_{11}

Fig. 4. Illustration of the Farthest First heuristic

5.5 Complexity of the algorithm

In this section we analyse our algorithm in term of the computational complexity (in the worst case), examining only FF heuristic, the others can be directly deduced.

It is easy to see that the active operation is to calculate the shortest arc-disjoint paths from l_i to f_0 (line 18 in Step 3.3), so we just focus on it and examine the computational times for it from the inside to the outside. First, the basic operation of it (compute the shortest path from a source to a single sink) takes $O(M' + N'logN')$ times in the worst case, where N', M' is the number of nodes and arcs of the remaining graph G', respectively. The *for* loop (for l_i) in the line 17 is repeated in maximal $O(|D|)$ times. Then the **for** loop to find T_0 in C_{max} (line 14), it takes $O(|D|)$ in the worst case. Finally, the **while** loop (line 9) also takes $O(|D|)$ times. In total, it takes $O(|D|)*O(|D|)*O(|D|)*O(M'+N'logN') = O(|D|^3 * (M' + N'logN'))$ times.

Let N, M be the number of nodes and arcs of the original graph G, respectively. We see that, N', M' is depended on N, M and D, the more D, the less N', M' compared with N, M, respectively. The fact that most practical optical core networks are sparse, with the degree of nodes on average of 3 or 4. Thus in this study, we just generate the graphs in which $M = k * N, k = \{3, 4\}$. Hence, $O(M) = O(N)$. Moreover, D is set such that $|D| \leq N/2$. Thus, we can assume that N', M' close to N, M, respectively. That means, the complexity can be $O(|D|^3 * (M' + N'logN')) = O(|D|^3 * (M + NlogN)) = O(|D|^3 * (NlogN))$ times at the worst case. However, in the general it is much better, because the **for** loop to find T_0 in C_{max} (line 14) takes $O(|D|)$ in the worst case, but T_0 can be rapidly found after $c << |D|$ times.

6 Experimental Results

In this section, we show the performance of our algorithm and compare with the other algorithms proposed in [1] (MDT) and [3] (Farthest Greedy and Nearest Greedy). In order to fairly compare with MDT in [1], the modified version of it is developed, namely Modified-MDT (MMDT for short) which is mentioned in the following.

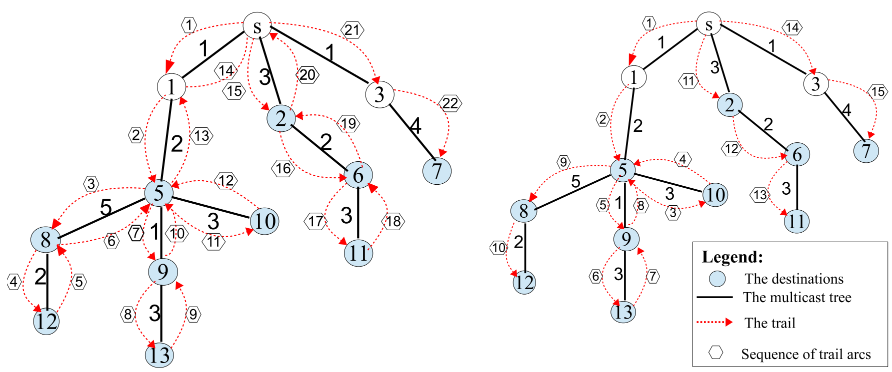
6.1 MDT and MMDT

As mentioned in Section 2, the MDT algorithm has two steps. The first step is to compute an approximated Steiner tree (AST) for a multicast request using the Minimum Cost Path Heuristic (MCPH) proposed in [7]. In the second step, a trail is computed based on the backtracking method following the AST. The backtracking phase starts from the root of the tree, and recursively repeats at each non-leaf node in the tree, say, the *current* node. In the downstream direction, the algorithm tries to include all the downstream links between the *current* node and all its children destinations. Backtracking is required when a leaf node is reached and there are still some destination nodes not yet visited.

However, the total cost and the diameter of the MDT trail are high because of multitude of round-trip traversing. Moreover, it is worth noting that, the source can inject the light signal by multiple transmitters on the same wavelength independently. By taking advantage of this feature, we developed the algorithm MMDT by modifying MDT in the backtracking phase, in such a way that it can eliminate the round-trip traversing the source while using only one wavelength.

The MMDT works as follows. First, it generates an AST using the MCPH just like the way of MDT. Then the backtracking method to each subtree of the AST (the nodes 1, 2 and 3 in Figure 5) is evoked, with a greedy sequence such that the trails growing to the nearest branch first (in term of cost of the branch). Consequently, there is no reversal arcs needed in the farthest branch for each sub-tree. Accordingly, the result is the set of trails rooted at the source, covering all the destinations with only one wavelength, but with multiple transmitters, one transmitter for each trail. Obviously, the diameter and the total cost of the resultant trials are less than those resulted by MDT.

To demonstrate MDT and MMDT, we use the same topology as the one shown in Figure 3 with a few changes: all the links are now bidirectional and the destination set is $D' = \{2, 5, 6, 7, 8, 9, 10, 11, 12, 13\}$. Figure 5 (a) demonstrates the computation of the multiple-destination trail according to the MDT algorithm and Figure 5 (b) demonstrates the MMDT for the same request $r = (s, D')$. As we can see, MMDT can reduce seven arcs compared with MDT, while both use only one wavelength.



(a) The MDT (b) The MMDT

Fig. 5. Illustration of MDT and MMDT

6.2 Two simulation settings and the performance metrics

Our algorithms can work in arbitrary directed graphs, meaning that unidirectional arcs and edges corresponding to two arcs on opposite directions can coexist in the graph, and the costs for the arcs can be given differently, even with arcs on opposite directions. However, the algorithms proposed in [1] and [3] supposed to work with bidirected graph, in which all the links are all bidirectional. Thus, for a fair comparison, we divided the simulations into two settings. In the first setting, all the algorithms are run on bidirected graphs, and in the second one, they are run on arbitrary directed ones.

Three performance metrics are taken into account in the simulations: the number of wavelengths required, the total cost and the diameter of the resultant routes (light-trails or light-forests). The diameter is defined as the number of maximal hop counts from the source to all the destinations. The reason for evaluation of this metric is that it can be represented for the end-to-end maximal delay. In fact, the delay can be combined by switching, queueing, transmission and propagation components. In all-optical networks, because of the high light speed, the propagation delay can be assumed to be the same on different links, and it is much less than the other components at the hops. Thus, the number of hops that the light signal has to pass by is usually used to represent for the delay.

6.3 Experimental results with bidirected graphs

In this setting, the considered algorithms have been run on several random bidirected graphs with different number of nodes $N = \{100, 200, 300\}$, the costs of arcs are randomly selected from the set of integer $\{1, 2, .., 20\}$, and the set of destinations D are also randomly selected with different size $|D| = \{10, 20, .., N/2\}$. To be sure that there is a feasible solution for all the algorithms, the selected graph must be connected and there is at least one directed path from the source to each destination for every simulation. Moreover, in order to guarantee a good confidence interval, for each size $|D|$, we run 100 simulations with different source and destination set. That means, each point in the resultant figures below is calculated on average of 100 (successful) simulations.

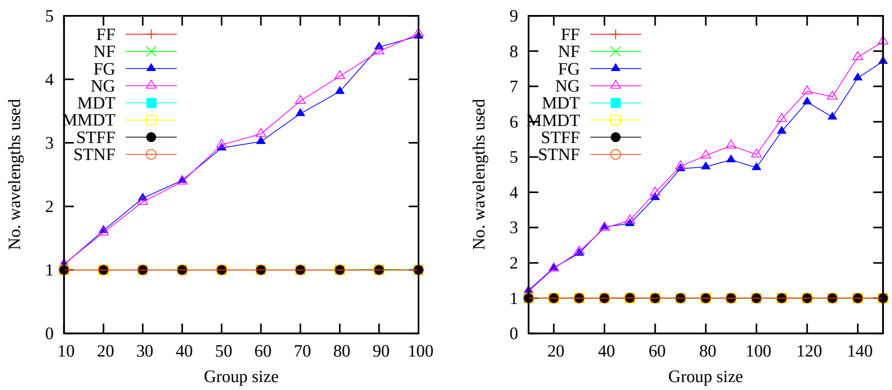
Besides, to respect the effect of the coefficient α on the performance of the proposed algorithms in [3] (FG and NG), we also set the coefficient α in {50, 100, 150}. The simulations showed that, in the cases of $\alpha = \{50, 100\}$ only FG and NG give slightly different results, in which the number of wavelengths is slightly higher and the total cost is slightly lower than those in the case of $\alpha = 150$. Thus we just show the results for the case of $\alpha = 150$. Likewise, we just show the results for the case of $N = \{200, 300\}$. In the case of $N = 100$, the results are quantitatively the same.

In Figure 6, FG and NG result in large number of wavelengths, ranging from 1 to 5 in 200 node-networks, and from 1 to 9 in 300 node-networks when the group size varies from 10 to $N/2$, while MDT and MMDT and the four variants of our algorithm draw a horizontal line with just 1 wavelength.

In Figure 7, the two variants of [1] appear with highest cost, then the two variants of [3], the four variants of our algorithm outperform the others. Among the four our algorithm variants, STFF has the lowest cost, and NF has the highest cost with a small difference.

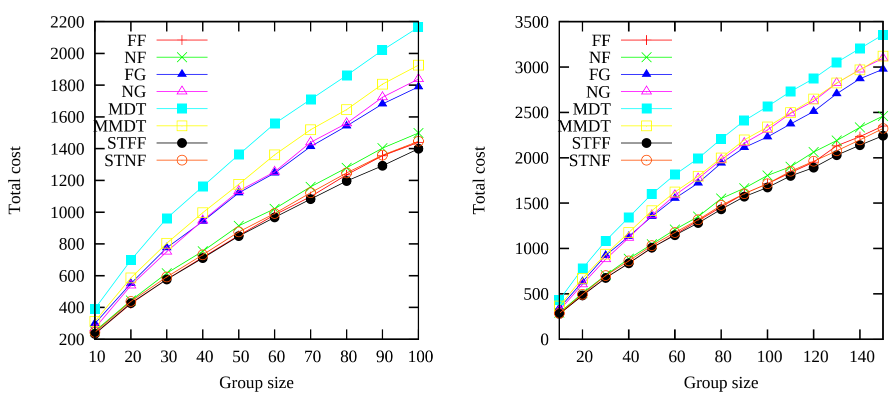
In Figure 8, about the diameter, unsurprisingly, the two variants of [1] appear with highest diameter, the two variants of [3] achieve a constant low diameter, the four variants of our algorithm are in the middle, with the lowest of NF and the highest of STFF.

In general, in bidirectional graphs, our algorithm result in the light-trail(s) in which the number of required wavelengths is close to 1, relatively low cost but quite high diameter compared with the other algorithms.



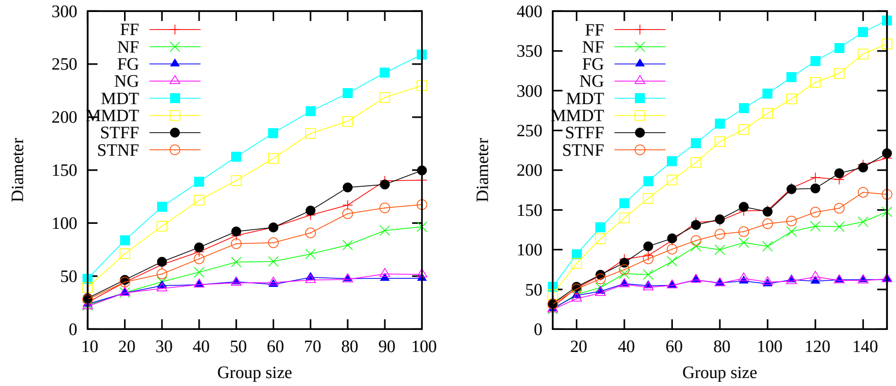
(a) N=200 (b) N=300

Fig. 6. No. Wavelengths vs. Group size



(a) N=200 (b) N=300

Fig. 7. Total Cost vs. Group size



(a) N=200 (b) N=300

Fig. 8. Diameter vs. Group size

6.4 Experimental results with directed graphs

The configurations of this setting are similar as the ones in the first setting, except that the graphs we work with are all arbitrary directed graphs. Similarly, we just show the figures for the case of $N = \{200, 300\}$, $\alpha = 150$ (cf. Figures 9, 10, and 11). In the other cases ($N = 100$, $\alpha = \{50, 100\}$) the results are quantitatively relative the same.

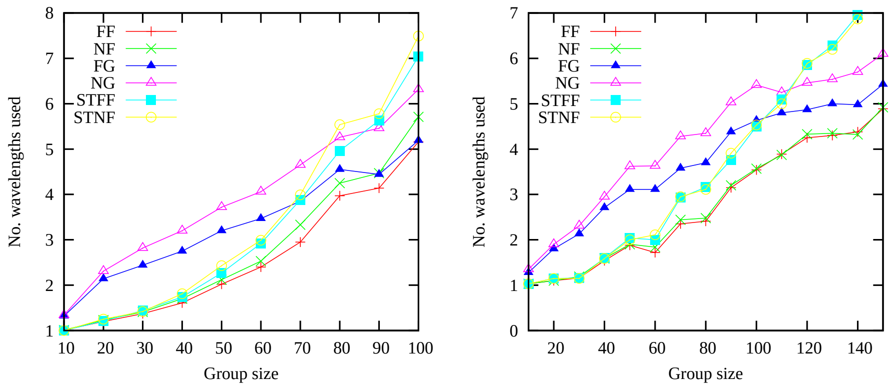
At first, in these configurations, MDT and MMDT almost have no solution so we do not show their results.

In Figure 9, all the algorithms result in the increasing number of wavelengths when the group size increases, and there is a difference between the variants according to the group size. When the group size is less than about 35%, FG and NG result in larger number of wavelengths with the largest of NG, while the four variants of out algorithm slowly increase, with the lowest of FF, and the highest of STNF. When the group size is lager than 40%, STFF and STNF increase faster and get over FG and NG; while FF and NF remain low, with the best of FF. In short, FF and NF provide a lowest number of wavelengths with a slightly difference between them, STFF and STNF perform pretty well with small group size, but do worse with large group size.

In Figure 10, all the algorithms appear with the same cost with a little dominance of FF.

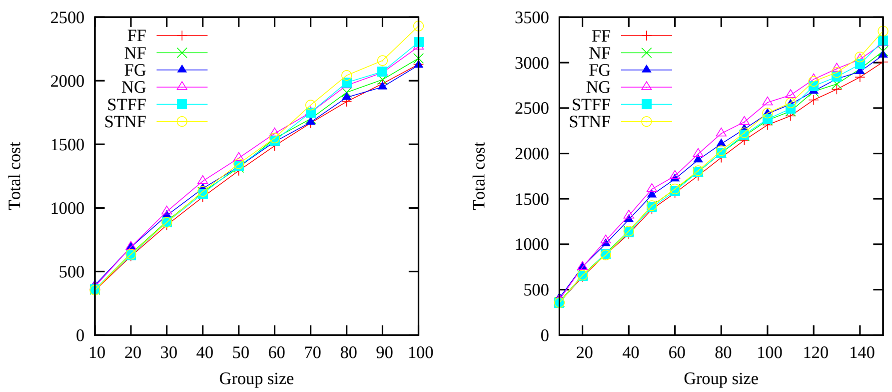
In Figure 11, again, the two variants of [3] achieve a constant low diameter with a slightly difference between them (lower than 25% of the number of nodes). Among variants of our algorithm, the two variants based on Nearest Greedy (NF, STNF) result in lower diameter (lower than 35% of the number of nodes). NF gets closer to them, especially almost the same when the group size gets closer 50%. The variants based on Farthest Greedy (FF, STFF) result in high diameter (around 50% of the number of nodes).

In short, in asymmetric directed graphs, our algorithms produce the light-trails with relatively low number of required wavelengths, a reasonable cost but quite high diameter compared with the other algorithms.



(a) N=200 (b) N=300

Fig. 9. No. Wavelengths vs. Group size



(a) N=200 (b) N=300

Fig. 10. Total Cost vs. Group size

6.5 Experimental result analysis

MDT versus MMDT

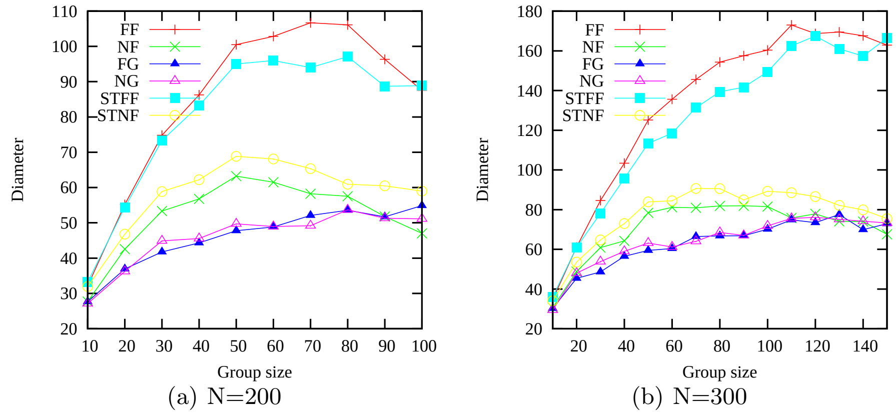


Fig. 11. Diameter vs. Group size

MDT and MMDT are just suitable in symmetric networks in which they always result in optimal wavelengths but high cost and diameter. By taking advantage of multiple transmitters, MMDT also needs just one wavelength (Figures 6) and it is even better than MDT in term of cost (Figures 7) and diameter (Figures 8). However, both have very poor performances in arbitrary directed graphs, even have no solution in most of the cases. This is because only one arc missing on the computing trail can make MDT and MMDT fail to get a solution.

Light-hierarchy based solution versus light-tree based solution

As seen in the experimental results above, light-hierarchy based solution (FF and NF) of our algorithm outperform light-tree based solution (FG and NG) [3] in term of the number of wavelengths and also the total cost, especially in bidirectional networks, at the expense of the diameter. This can be explained as follows.

The two approaches start with the same DSPT tree. In the rerouting phase, FG and NG try to extend the tree but always keep the tree structure which does not allow any cycles. Moreover, the nodes used for the tree extension are always restricted. They are the source or the leaves which are either the farthest (FG) or nearest (NG) destinations in each subtree of the computed trees. These two properties restrict the number of destinations covered in one tree, inducing lager number of wavelengths needed and higher cost. Besides, cycles are not allowed in the forest, and when the destinations cannot be routed in the current tree, they are routed in a new tree by the shortest paths, so the diameter is short.

In contrast, our approach is more flexible. After the first step, the structure is no longer a tree, but a set of trails (composing a hierarchy) which allow cycles without conflict between them. With this property, more arcs can be used for the trail extension. Moreover, the nodes used for the trail extension are not restricted. All of the terminals of the existed trails (and the source) are taken into consideration to select the best one (in term of cost). In other words, more nodes can be used for the trail extension. These two properties increase the number of destinations can be covered in one trail, resulting in a small number of wavelengths required, but, of course, with a longer diameter. Besides, because more nodes are considered for the best cost, a lower total cost of the final trails can be achieved.

Farthest First versus Nearest First

As shown in the experimental results above, FF results in higher diameter but a slightly fewer wavelengths and a quasi-similar cost compared with NF. This can be explained as follows.

First, the diameter is the number of hops (the length) of the longest trail. When the selected trail (the *routed trail*) is replaced by an other one (the *routing trail*), the new routing trail must be longer (in term of cost) than the routed trail. Thus, the longer the routed trail is, the longer the new length can be. FF chooses the longest trail in the maximal clique, hence it makes the new trail longer than NF does. Furthermore, because the new trail is usually longer than routed trail in the old maximal clique, and it will probably become the farthest one in the new maximal clique and will be first considered next time. Hence it becomes longer and longer, and finally it can correspond to the diameter of the final trails. That is the reason for the fact that FF results in a longer diameter than NF.

Similarly, since FF tends to include more destinations in a long trail, the probability that the number of wavelengths that can be reduced by FF is higher than by NF. So FF results in a fewer wavelengths than NF does.

Finally, when the routed trail is replaced, the reduced cost is calculated by the cost of the routed trail minus the cost of the extended path of the routing trail (or the total cost of the routing trail if the source is selected). Thus, the longer the selected trail is, probably that the more the reduced cost can be. Since FF chooses the longest trail and NF chooses the shortest one in the maximal clique to diminish first, FF can reduce more cost than NF.

DSPT-based solution versus DAST-based solution

As shown in Figure 9, DAST-based solutions (STFF and STNF) result in larger number of wavelengths compared with DSPT-based solutions (FF and NF), even larger than FG and NG at large group size. This can be explained as follows. The DSPT-based solutions started with a DSPT which tends to create more branches composing a star surrounding the source. Because the source can be equipped with multiple transmitters, so it can serve as a branching node with arbitrary degree. Whereas, a DAST tends to include more destinations in fewer branches. Thus the DAST does not take better advantage of the source. Moreover, a DSPT probably produces more terminals (leaves) than a DAST. Consequently, when the group size is large, a DSPT creates more chances for the terminals to reroute than a DAST, leading that more wavelengths can be reduced with DSPT-based solutions than DAST-based solutions.

7 Conclusion and Future works

In this paper we address the multicasting problem in all-optical networks without splitters. The problem is to find a set of light-trails which minimizes the number of required wavelengths with a low cost. The problem is proved to be NP-hard even in symmetric networks, and four heuristics are proposed to make a feasible solution, two are Shortest Path Tree Based, and the others are Approximated Steiner Tree Based. The idea of our algorithm is to diminish the conflict between the light-trails until it cannot be reduced. Especially, unlike the popular approaches which assume to work in symmetric networks, our algorithm can work well in arbitrary networks.

The four heuristics of our algorithm are compared with the proposed algorithms in the literature, and the simulation results showed that our algorithm achieves low number of used wavelengths, low cost but quite high diameter. The experimental results also showed that the solutions based on DSPT are better than those based on DAST in term of the number of wavelengths used, the cost and the diameter of the trails. Among the four proposed heuristics, although the Farthest First can result in smaller number of wavelengths and lower cost, the Nearest First provides a better trade-off among the three performance metrics.

However, this study has some limitations. First, it is worth to show how good our heuristic algorithm is in comparison with the optimal solution. Thus, we will formulate the exact solution for this problem and make the comparison in the next works.

Moreover, in this paper we just deal with the networks in which all the links are assumed to have the same available wavelengths. To be more realistic, the distribution of wavelengths in the network links can be arbitrary, depending on the state of the networks where many requests are on-going together. In this case, each wavelength corresponds to a topology graph that is different from another. The routing problem can be more complicated but more interesting.

References

1. M. Ali and J. S Deogun. Cost-effective implementation of multicasting in wavelength-routed networks. *IEEE/OSA Journal of Lightwave Technology*, 18:16281638, 2000.
2. M. Ali and J. S Deogun. Power-efficient design of multicast wavelength routed networks. *IEEE Journal on Selected Areas in Communications*, 18:1852–1862, 2000.
3. Der-Rong Din. Heuristic Algorithms for Finding Light-Forest of Multicast Routing on WDM Network. *Information Science and Engineering*, 25:83–103, 2009.
4. Steven Fortune, John Hopcroft, and James Wyllie. The directed subgraph homeomorphism problem. *Theoretical Computer Science*, 10(2):111 – 121, 1980.

5. Deying Li, Xiufeng Du, Xiaodong Hu, Lu Ruan, and Xiaohua Jia. Minimizing number of wavelengths in multicast routing trees in WDM networks. *Networks*, 35(4):260–265, 2000.
6. Miklos Molnar. Hierarchies to Solve Constrained Connected Spanning Problems. Technical Report 11029, LIRMM, September 2011.
7. H. Takahashi and A. Matsuyama. An approximate solution for the Steiner problem in graphs. *Mathematica Japonica*, 24:573–577, 1980.
8. Yingyu Wan and Weifa Liang. On the minimum number of wavelengths in multicast trees in WDM networks. *Networks*, 45(1):42–48, 2005.
9. Xijun Zhang, John Wei, and Chunming Qiao. Constrained multicast routing in WDM networks with sparse light splitting. *IEEE/OSA Journal of Lightwave Technology*, 18:1917–1927, 2000.
10. Fen Zhou, Miklos Molnar, and Bernard Cousin. Light-Hierarchy: The Optimal Structure for Multicast Routing in WDM Mesh Networks. In *Computers and Communications (ISCC), 2010 IEEE Symposium on The 15th IEEE Symposium on Computers and Communications (ISCC2010), 2010*, pages 611 – 616, Riccione Italie, June 2010.
11. Yinzhu Zhou and Gee-Swee Poo. Optical multicast over wavelength-routed WDM networks: A survey. *Optical Switching and Networking*, 2(3):176 – 197, 2005.