Multicast Routing in WDM Networks without Splitters

Dinh Danh Le, Miklos Molnar, Jérôme Palaysi

Multicast Routing in WDM Networks 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 directed light-trails. The problem is NP-hard even in symmetric digraphs. An efficient heuristic is proposed to minimize the conflicts between the light-trails, and so to minimize the number of used wavelengths. The performance is compared to existing light-trail based heuristics and the four our algorithms provide 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 promising 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 are present in networks to efficiently perform data transmission from a source to several destinations. Usually, multicast routes corresponds to trees in the topology graph. To perform multicast, there should 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. In fact, splitters are expensive and the light power can be decreased considerably by splitting (it can be inversely proportional with the number of outgoing ports [2]). This constraint prevents all-optical multicasting from employing splitters.

In our paper, we investigate an 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 wavelengths. Our objective is to perform multicasting without splitters and minimizing the number of used wavelengths. Solutions in bidirectional networks (where a wavelength is available in both directions between the connected switches) are known, but we investigate the arbitrarily directed case which is very practical. Even if the network is designed to be bidirectional, when some demands hold some of the resources of the network, the resulting network graph is now arbitrarily directed, therefore the routing for subsequent demands will be calculated on a digraph.

Some studies indicated that non-simple light-trails (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 wavelengths is a set of (non-simple) light-trails. The computation of the optimum in directed graphs is NP-hard. So, we propose some heuristic algorithms, which try to minimize the number of wavelengths, taking into account the availability of fibers in the network, with a low cost. We compare the performance of them with two previously proposed multicast routing algorithms (one based on light-tree [3], the other based on light-trail [1]).

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 an arbitrary directed graph (or digraph) $G = (V, A)$, in which each arc represents the availability of a fiber between the pair of nodes and there are at most two fibers between any pairs. This configuration is realistic in real optical networks. We suppose that each fiber has the same set of available wavelengths and each arc $e \in A$ is associated with 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. We also suppose that there is at least one directed path from the source to each destination, so there is a feasible solution for each request.

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: $TotalCost(T) = \sum_{i \in [1,k]} \sum_{e \in E(t_i)} cost(e)$.

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 multicast routing is equal to the maximum number of wavelengths that are assigned for one fiber for the given request.

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 D of destination nodes
Solution: a set T of light-trails originated from 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 D of destination nodes
Solution: a set T of light-trails originated from s and covering all the destinations
Objective: minimize the total cost of T

The solution of Problem 1 can be composed from very long trails. The optimum of Problem 2 can use a high number of wavelengths. Thus, the trade-off between the two 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 D of destination nodes, the number of wavelengths $W \in \mathbb{Z}^+$
Solution:a set T of light-trails originated from 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).
Instance: a network G, a source node s and a set D of destination nodes, the number $L \in \mathbb{Z}^+$
Solution: a set T of light-trails originated from s and covering all the destinations such that its length (the maximum number of hop counts from s to the terminal d_i) does not excess L
Objective: minimize the number of wavelengths used by T

[1] *Distinct wavelength constraint:* Different light-paths or light-trees sharing a common link must be allocated distinct wavelengths [7].

Notice that the optimal solutions for all the above problems (with the given constraints) corresponds to a hierarchy obtained from a star (cf. [6] for the definition of a hierarchy).

The mentioned routing problems are hard optimization problems. Problem 1 corresponds to finding the solution with minimum 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 to the Degree Constrained Directed Minimum Spanning Tree Problem in the *distance graph*[2] of the corresponding original graph.

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 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 and a request (s, D)
Question: Is there a spanning trail in G originated 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: Are there two paths P (between x and x') and Q (between y and y') in G that are arc-disjoint?

Since Problem 2ADP is the case of the Subgraph Homomorphism Problem when the pattern of two disjoint paths which is proved to be NP-complete [4], so 2ADP is NP-complete. In the following, we transform Problem 2ADP to Problem OST in order to prove that OST is also NP-complete.

Let G_2 and two pairs of vertices (x, x') and (y, y') be any given instance I of Problem 2ADP, we create a graph G_1 by 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_1 and the request (s, D) form an instance I' of Problem OST (Fig. 1).

[2] *distance graph* is a graph $G_d = (V_d, E_d)$ in which $V_d = \{s, d_i(i = 1, .., k)\}$, and each edge $e \in E_d$ corresponds to a shortest path between the two end-vertices of e computed in its original graph.

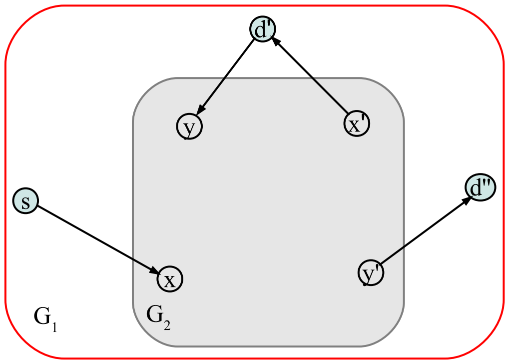


Fig. 1. Illustration of the two problems OST and 2ADP

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 $x_0 = x, x_k = x', y_0 = y, y_l = y'$, are a solution S for Problem 2ADP. Let S' be the trail $(s, x_0, x_1, ..., x_k, d', y_0, y_1, .., y_l, d'')$ (Figure 1). Obviously, there is no arc in S' occurring more than once, so S' is a trail (and thus, a solution) for the Problem OST.
2. In contrast, suppose that there exists a solution (i.e., a trail) S' for Problem OST. S' is of the form: $(s, x_0, x_1, ..., x_k, d', y_0, y_1, .., y_l, d'')$ where $x_0 = x, x_k = x', y_0 = y, y_l = y'$. Let $P = (x_0, x_1, ..., x_k), Q = (y_0, y_1, .., y_l)$. Because S' is a trail, all the arcs occur once. So the two paths P and Q are arc-disjoint. Thus, P and Q form a solution S for Problem 2ADP. ■

Because 2ADP is NP-complete, OST is NP-complete. The Problem P_1 is to minimize the number of wavelengths used, i.e., minimize the number of colors needed to assign the trails such that shared-arc trails must be assigned with different colors. In OST we just consider the solution with one trail (equivalent to one color needed), however, it is easy to verify that the problem with multiple trails using one color oriented from the source covering all destinations is also NP-complete (we call it One Color Multiple Spanning Trails-OCMST). The Problem OCMST is the special case of the more general Problem P_1 (P_1 asks for a minimum number of colors), so Problem P_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. [10] for some basic algorithms and [12] 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 [9].

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 total cost of the edges traversed. 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 be equipped with at least two fibers in order to support bidirectional transmission on the same link. 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 computing an approximated Steiner tree for a multicast request using the Minimum Cost Path Heuristic (MCPH) proposed in [8]. A trail is then 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 is required in both directions, hence the total cost and the diameter of the light-trail is always 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 number of arcs (that backtrack 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 6.

In [3], Der-Rong Din posed the Minimal Cost Routing Problem which minimizes the cost under WDM symmetric networks using only TaC cross-connects. Unlike the approach of [1] that based on light-trail, the approach of Der-Rong Din is based on *light-forest* (a set of the *light-trees* [12]), rooted at the source and covering all the destinations. Besides, the source can inject the signal by multiple transmitters so that each light-tree can use a single wavelength. Furthermore, to produce a trade-off between the total cost of the light-forest and the number of wavelengths used, the author developed an objective function which combines the actual total cost of the light-forest and the cost for using wavelengths: $f = cost(F) + \alpha * numWL$, in which F is the resultant light-forest, $numWL$ is the number of wavelengths used, and α is a specific coefficient.

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 as follows. FG keeps the farthest (in term of cost) destination routed by the computed shortest path, and chooses the farthest destination in the unreached set to reroute, whereas NDF keeps the nearest destination and chooses the nearest destination in unreached set to reroute in the rerouting phase. The rerouting phase is performed by the shortest paths from the source or from the leaves of computed trees to each unreached destination, such that the paths 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 is better than NG and MDT.

Most of 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 allows cycles but they are only 2-cycles[3]. In fact, one can construct the routing structures that allow not only 2-cycles but also arbitrary ones. These structures correspond to a *hierarchy* that was proposed in [6]. For multicast routing in WDM networks, the *light-hierarchy* concept (i.e., a hierarchy using a single wavelength) has been proposed in [11]. In this study, we try to create advantageous hierarchies corresponding to 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 Approximated 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's algorithm) ensuring that the shortest paths are loop-free. In contrast, a DAST can be computed by employing a Steiner heuristic, e.g., the MCPH proposed in [8]. In our algorithm, one of these

[3] An n-cycle is a cycle with n vertices.

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., two trails share a common arc.
In this study, because the trails initially are the directed paths from the root to the leaves of the rooted directed tree, the conflicts only occur in the prefixes of the concerned trails. This property is preserved during trails manipulation in our proposed algorithm described in following sections. Accordingly, the following properties are introduced.

Property 1: Each connected component in the conflict graph G composes a (conflict) clique[4] of G.
Indeed, each connected component in the conflict graph corresponds to a subtree of the root of the DST (cf. Figure 2). All the trails in the same connected component share the first arc from the source, so they conflict one another. Thus Property 1 follows.

Property 2: 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 cardinality or the size) of the maximum clique in the conflict graph.
As mentioned, each clique corresponds to a subtree of the root of the DST. These subtrees are arc-disjoint, so the corresponding trails in each clique do not share any arcs with those 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 cardinality of the maximum clique, because we can use some colors that have been used in the maximum clique to re-assign the 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 the clique. So this property holds.

Thus, the problem of minimizing the number of used wavelengths reduces to the problem of minimizing the cardinality of the maximum clique in the conflict graph.

First destination: Let T_i be the considered trail, and l_i is its terminal. Let e be the first shared arc counted from l_i to the source. We define f_i the first destination of T_i if f_i is the first destination on the path from the target node of e to l_i. If there is no other destination but l_i, we consider f_i as l_i by default. In our study, the first destination is used to diminish the trails from the maximum clique, so that the number of wavelengths can be reduced.

[4] A clique of a graph is a complete subgraph of the graph.

Arc-disjoint paths: The paths that do not share any arcs. In our study, only arc-disjoint paths are used to diminish the number of conflict trails. The explanation with Figure 3 is presented below.

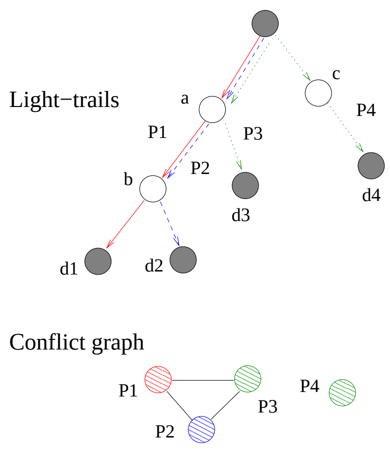


Fig. 2. Example of a DST and its conflict graph

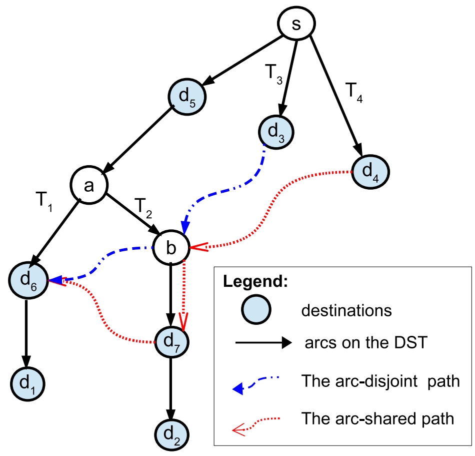


Fig. 3. Illustration of first destination and arc-disjoint paths

Figure 2 illustrates a set of paths (trails) composing a DST for the multicast request $r = (s, \{d_1, d_2, d_3, d_4\})$ and the corresponding conflict graph. There are two cliques corresponding to two subtrees of the DST. The maximum clique is composed from the paths T_1, T_2, T_3 orienting to the destinations d_1, d_2, d_3, respectively. Three colors are needed for these three trails. The other clique composed only from the path T_4 that can re-use one color that was assigned for the maximum clique.

Figure 3 illustrates a DST originated from the source s spanning the destination set $\{d_1, d_2, d_3, d_4, d_5\}$. There are four initial trails (paths) T_1, T_2, T_3, T_4 orienting to the destinations d_1, d_2, d_3, d_4, respectively. Now we calculate the first destinations for the two trails T_1 and T_2. We see that T_1 and T_2 share the two arcs (s, d_5) and (d_5, a), in which (d_5, a) is the first shared arc between the two trails (counted from the terminals) and a is the target node of it. Because d_6 is the first destination on the path from a to d_1, according to the definition, d_6 is the first destination of T_1. Besides, there is no destination from a to d_2 (except d_2), so the first destination of T_2 is its terminal d_2.

Also in Figure 3, there are two paths orienting to the first destination d_6 of T_1: the first one (d_3, b, d_6) is an arc-disjoint path and the other (d_4, b, d_7, d_6) is an arc-shared path which shares the arc (b, d_7) with the existing trail T_2. Only arc-disjoint paths are valid for rerouting the trails in our algorithm. This is because if the arc-shared path is used, although one trail will be removed, the new conflict will be created, thus the conflicts still remains. In deed, if the path (d_4, b, d_7, d_6) is used, although the trail T_1 will be removed, the new conflict is created by the new trail $(s, d_4, b, d_7, d_6, d_1)$ sharing arc (b, d_7) with T_2, so the number of colors is not reduced.

5 Proposed Heuristics to Minimize the Wavelengths

In this section we present our proposed algorithms. We begin with the general idea of our algorithm, then four possible heuristics are briefly described in which one is chosen to detail. The illustration and the computational complexity of the algorithm are also given.

5.1 The Algorithm Framework

The idea of the algorithm is to diminish the number of nodes (trails) in the maximum 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 iteratively tries to diminish the number of trails in the maximum clique, says C_{max}. At each step, it selects one trail from the maximum clique that can be replaced by another trail without conflict, says T_0. Some mechanism can be employed to select this trail (see propositions later). 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 there is an *arc-disjoint shortest* path from the terminal of T_k to the *first destination* of the trail T_0. Then T_0 is replaced by T_k, and the cardinality of the C_{max} is reduced by 1. The algorithm iterates until the maximum clique cannot be reduced.

The framework of the algorithm consists of four main steps that can be described as follows. For the sake of convenience, for every trail T_i, we denote l_i

the terminal and f_i the first destination of it; also, we denote P_{u-v} the path from node u to node v.

THE ALGORITHM FRAMEWORK

Step 1: Compute a Directed spanning tree (DST) from the source s covering all destinations. If there are no branching nodes in the DST, then DONE.
Step 2: Compute the conflict graph from the DST.
Step 3: Repeat Step (3.1) to Step (3.4) in the following until the cardinality of the maximum clique C_{max} cannot be reduced.
Step 3.1: Find the maximum clique C_{max}[5].
Step 3.2: Select a trail T_0 in C_{max}.
Calculate the terminal l_0 and the first destination f_0 of T_0.
Step 3.3: Calculate the arc-disjoint shortest path P_0 from the source s to f_0 (if any).
For every trail T_i (except T_0), compute the path P_i from l_i to f_0 such that it is arc-disjoint (with all the current trails) and the shortest one.
Calculate P_k as the shortest path among the paths $P_i (i \geq 0)$.
If P_k is not found, go to Step 3.2 to select another trail T_0.
If P_k is not found for all trail T_0 in C_{max}, go to Step 4 to finish.
Step 3.4: If $P_k = P_0$, create a new trail $T'_0 = P_0 + P_{f_0 - l_0}$.
Otherwise, create a new trail $T'_k = T_k + P_k + P_{f_0 - l_0}$.
Replace T_0 by T'_k, reduce the cardinality of clique C_{max} ($|C_{max}|$) by 1.
Step 4: Record $|C_{max}|$ as the minimum number of wavelengths required, return the final trails.
Employ the trail-wavelength-assignment (TWA) algorithm (described below) to assign wavelengths for the set of final trails.

5.2 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 one trail of every clique in the set of the remaining cliques. Repeat that until there is no trail in the set of remaining cliques.

5.3 Complexity of the algorithm

In this subsection we analyse our algorithm in term of the computational complexity (in the worst case). Let N, M be the number of nodes and arcs of the

[5] 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 put into the queue.

original graph G, respectively; D be the destination set. The fact that most of practical optical core networks are sparse, with the degree of nodes on average of 3 or 4. Thus in this study, we just deal with the graphs in which $M = k * N, k = \{3, 4\}$.

In Step 1, if DSPT is used for DST, computed by using Dijkstra's algorithm with heap data structure, it takes $O(M+NlogN)$ times. Because $M = k*N, k = \{3, 4\}$, it is equal to $O(NlogN)$ times. If DAST is used for DST, computed by using MCPH, it takes $O(|D|N^2)$ times [8].

Step 2 takes $O(|D|)$ times.

Step 3 repeated at most $|D|$ times.

Step 3.1 takes at most $O(|D|)$ times.

Step 3.2 takes at most $O(|D|)$ times.

In Step 3.3, it is easy to see that the dominant operation is to calculate the shortest arc-disjoint paths from l_i to f_0, it takes $O(NlogN)$ times, and it is repeated for all the existing trails, i.e., at most $|D|$ times. Thus, this step takes at most $O(|D|NlogN)$ times.

Step 3.4 takes $O(1)$ time.

Note that Step 3.2 and Step 3.3 are repeated at most $|D|$ times (to select T_0 in C_{max}) with the domination of Step 3.3. So, the complexity of these steps is $O(|D| * |D| * NlogN) = O(|D|^2 NlogN)$.

Step 3.3 is also dominant all the steps in Step 3, so Step 3 takes $O(|D| * |D|^2 NlogN) = O(|D|^3 NlogN)$

In Step 4, the TWA is dominant, it takes $O(W)$ with W being the number of given wavelengths.

On the whole, Step 1 and Step 3 dominate the others in which the complexity of Step 1 depends on which kind of DST is used. Thus, if DSPT is used, the complexity of algorithm is $O(NlogN) + O(|D|^3 NlogN) = O(|D|^3 NlogN)$. If DAST is used, the complexity of algorithm is $O(|D|N^2) + O(|D|^3 NlogN) = O(|D|N(N + |D|^2 logN))$.

However, in most of cases, the complexity can be much better for the following reasons. First, the selection of T_0 in steps 3.2-3.3 can be done in a few times (because whenever the path P_k is found, it quits the loop). Second, in Step 3, we only consider the cliques containing more than one node (trail), i.e., Step 3 is repeated less than $|D|$ times. Finally, in Step 3.3, whenever P_k is not found for all existing trails, the algorithm is about to stop, although there are several cliques remaining. That means, the steps 3.2-3.3 are repeated just c times with $c < |D|$ being the size of current maximum clique.

5.4 Two trail selections

In step 3.2, a greedy mechanism is used to select the first trail T_0 in the maximum clique C_{max}. For this, we propose two different algorithms. The first one is Farthest First (FF) if T_0 is the trail containing the farthest terminal among the trails in C_{max}. The second one is Nearest First (NF) if the the trail containing the nearest terminal the trails in C_{max}. The descriptions of them are shown in the following.

5.5 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 the number of wavelengths.

construct a DSPT from s covering all destinations in D.
let T be the set of paths from s to every leaf of DSPT. If there are no shared-arc paths in T, then DONE.
calculate cliques $C_i, i = 1, 2, .., n_c$, n_c is the number of sub-trees of s in the DSPT.
put C_i with $|C_i| > 1$ into a priority queue $PQ = \{(C_i, |C_i|)\}$[6]
$G' \leftarrow G \setminus T$ $\{G'$ is the remaining graph by removing arcs corresponding to all the trails in $T\}$
while $PQ \neq \emptyset$ **do**
  pop the maximum clique C_{max} from PQ
  sort the trails in C_{max} in the descending order of their costs
  for each trail T_0 in the sorted C_{max} **do**
    find the terminal l_0 and the first destination f_0 of T_0
    calculate the arc-disjoint shortest path P_0 from the source s to f_0 in G'
    if P_0 is found **then** $P_k \leftarrow P_0$
    else $P_k \leftarrow \emptyset$; $cost(P_k) \leftarrow \infty$
    for every $T_i (i \geq 1)$ **do**
      calculate the arc-disjoint shortest path P_i from the terminal l_i to f_0 in G'[7]
      if $cost(P_i) < cost(P_k)$ **then** $P_k \leftarrow P_i$
    end for
    if P_k is found **then break**
  end for
  if P_k is not found **then break**
  if $P_k = P_0$ **then** $T_{new} \leftarrow P_0 + P_{f0-l0}$; $T \leftarrow T \cup \{T_{new}\}$
  else $T_k \leftarrow T_k + P_k + P_{f_0-l_0}$; $T \leftarrow T \cup \{T_k\}$
  $T \leftarrow T \setminus \{T_0\}$; $G' \leftarrow G \setminus T$; $|C_{max}| = |C_{max}|$ - 1.
  if $|C_{max}| = 1$ **then break**
  else update the new size $|C_{max}|$ for the C_{max} in PQ
end while
if $|C_{max}| > W$, **then return** FALSE
else record $|C_{max}|$ as the minimum number of wavelengths required
return T as the set of final trails
employ the TWA algorithm to assign wavelengths for T.

[6] In the priority queue, the larger the size of C_i, the higher the priority of it.
[7] In order to calculate the arc-disjoint path from s or l_i to f_0, the sub-path from the target node of the first shared-arc to f_0 is temporarily removed

5.6 The other heuristics

In the line 8 of FF's description, if the trails in the C_{max} are sorted in the ascending (instead of descending) order of their costs, we have the description of Nearest First heuristic.

If we use DAST instead of DSPT in line 1 of FF's description, we have another heuristic called Steiner tree Farthest First (STFF). In the same way, if we use DAST for NF, we have Steiner tree Nearest First (STNF).

In summary, we have four heuristics originated from our framework algorithm: two are DSPT based (FF, NF) and other two are DAST based (STFF, STNF). Their performances are evaluated and compared each other in the next section.

5.7 Algorithm illustration

In order to demonstrate the algorithm, we use a network in Figure 4 as an example. Moreover, to simplify, and because the other heuristics have the same principle, we just illustrate the heuristic FF in the Figure 5 below.

After the Step 1 and Step 2, the DSPT and the initial conflict graph are shown in Figure 5 a). The maximum clique comprises three paths T_{10}, T_{12}, T_{13}, in which T_{12} containing the farthest terminal, so it is selected first. The first destination f_0 of T_{12} is node 8, the arc-disjoint shortest path is the path passing the nodes {10, 5, 8}. Thus T_{12} is replaced by T_{10}, the new trail is $T_{12}^{'}$ (Figure 5 b)). Similarly, $T_{12}^{'}$ is then replaced by $T_{12}^{''}$ in the next run (Figure 5 c)). Finally, the final set of trails is obtained after T_{11} replaced by $T_{11}^{'}$ (Figure 5 d)). As we can see, only one wavelength now is enough to perform the routing.

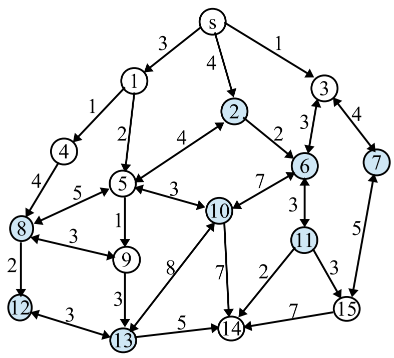
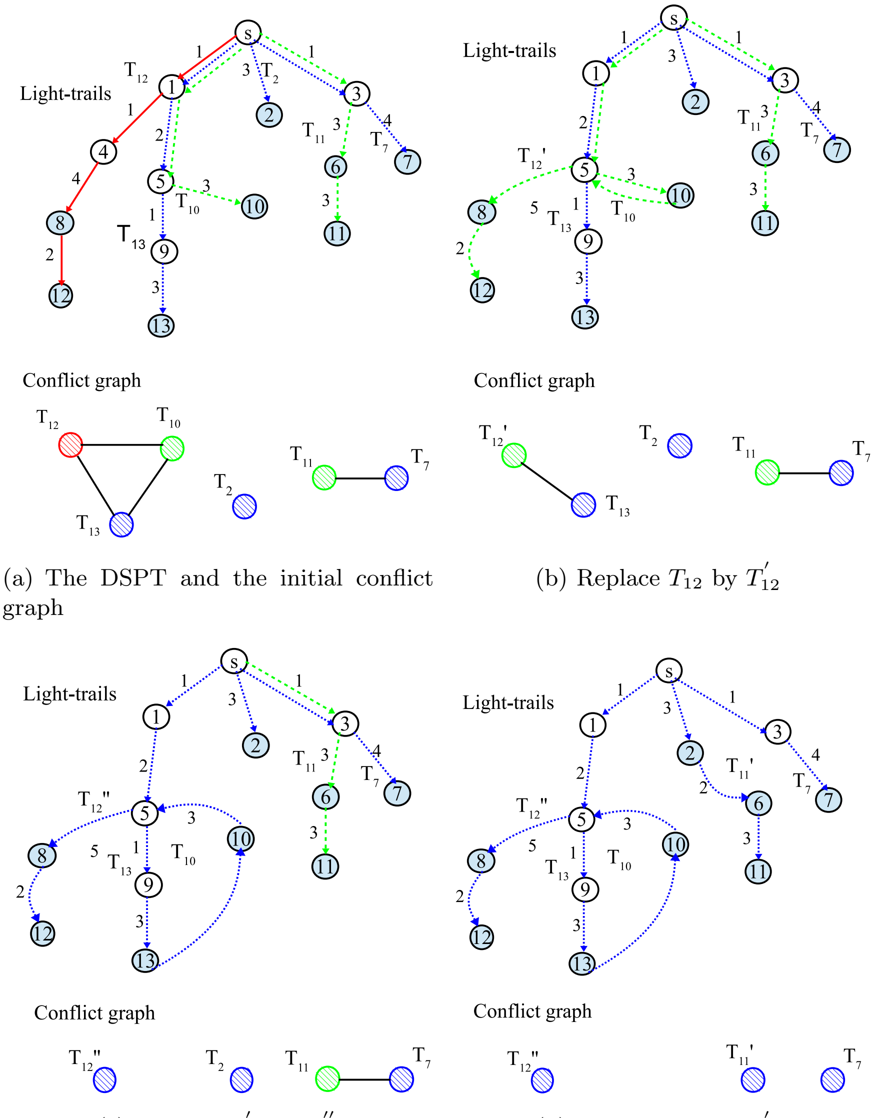


Fig. 4. A digraph to consider



(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. 5. Illustration of the Farthest First heuristic

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 heuristic in [1], the modified version of it is developed, namely Modified-MDT heuristic (MMDT for short) which is described in the following.

6.1 MDT and MMDT heuristics

As mentioned in Section 2, the MDT heuristic has two steps. The first step computes an approximated Steiner tree (AST) for a multicast request using the MCPH proposed in [8]. 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.

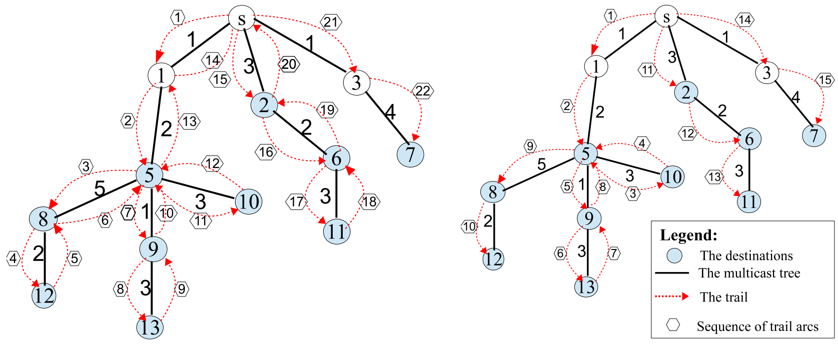
However, the total cost and the diameter of the MDT are high because of multitude of round-trip traversing. Moreover, it is worth noting that, the source can inject the light signal by multiple transmitters on the same wavelength independently. By taking advantage of this feature, we developed the MMDT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

The MMDT heuristic works as follows. First, it generates an AST using the MCPH just like the way of MDT heuristic. Then the backtracking method to each subtree of the AST (the nodes 1, 2 and 3 in Figure 6) 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 are 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 trails are less than those resulted by MDT heuristic.

To illustrate MDT and MMDT, we use the same topology as the one shown in Figure 4 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 6 (a) illustrates the computation of the multiple-destination trail according to the MDT heuristic and Figure 6 (b) illustrates the MMDT for the same request $r = (s, D')$. As we can see, seven arcs are reduced in MMDT compared with MDT, while both solutions use only one wavelength.

6.2 Two simulation settings and the performance metrics

Our algorithms can work in arbitrary directed graphs, meaning that unidirectional and bidirectional links 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 bidirectional. Thus, for comparison, we divide 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.



(a) The MDT (b) The MMDT

Fig. 6. Illustration of MDT and MMDT

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 maximum 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 maximum delay. In fact, the delay can be combined by switching, 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 $N = \{100, 200, 300\}$, with the average nodal degree of 4 (that is common in most of core optical networks today); 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 request. 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 FG and NG (cf. Section 3), we also set $\alpha = \{50, 100, 150\}$. 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 higher and the total cost is 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 7, 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 8, 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 9, 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 four algorithms result in few number of required wavelengths (close to 1), relatively low cost but quite high diameter in comparison to the other algorithms.

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 10, 11, and 12). In the other cases ($N = 100$, $\alpha = \{50, 100\}$) the results are quantitatively relative the same.

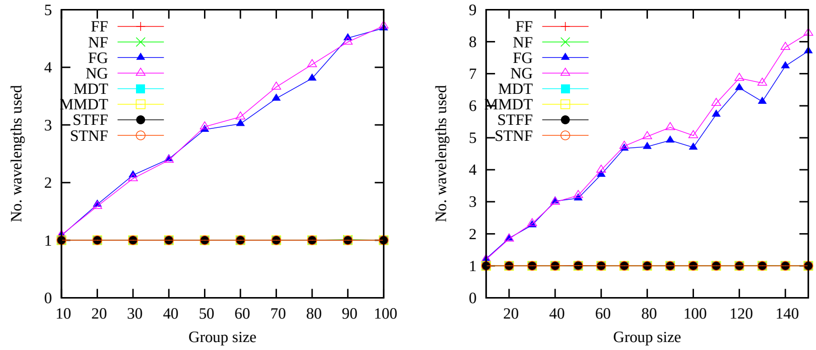
At first, in these configurations, MDT heuristic and MMDT heuristic cannot guarantee a solution so we do not show their results.

In Figure 10, all the algorithms result in the increasing number of wavelengths when the group size increases. When the group size is less than about 35%, FG and NG produce larger number of wavelengths with the largest of NG, while our four heuristics 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 quite well with a small group size, but do worse with a large group size.

In Figure 11, all the algorithms appear with the same cost with a little dominance of FF.

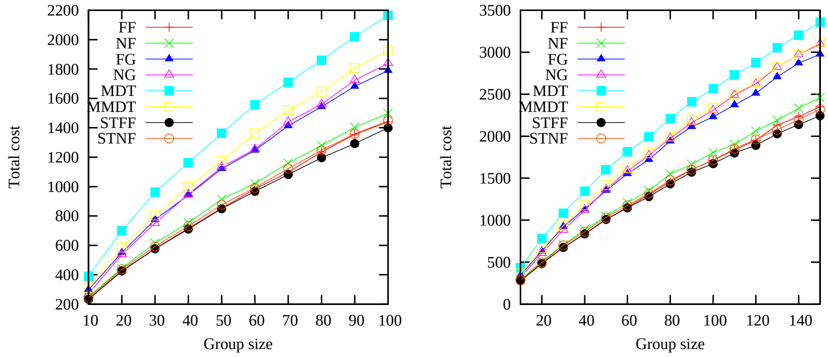
In Figure 12, again, the two variants of [3] achieve a constant low diameter with a slightly difference between them (lower than 25% of the number of nodes). Among variants of our algorithm, the two variants based on Nearest Greedy (NF, STNF) result in lower diameter (lower than 35% of the number of 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.



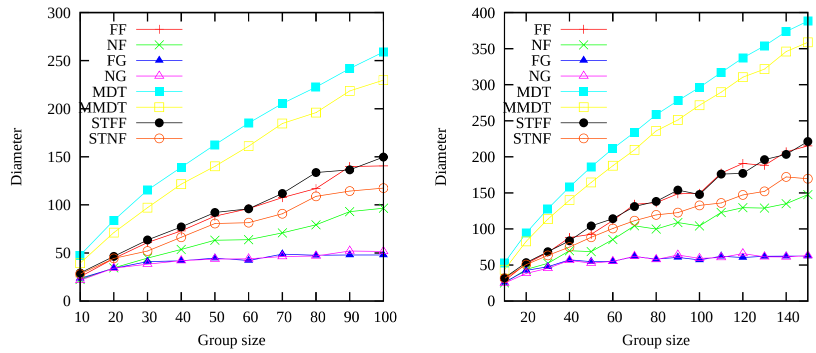
(a) N=200 (b) N=300

Fig. 7. No. Wavelengths vs. Group size



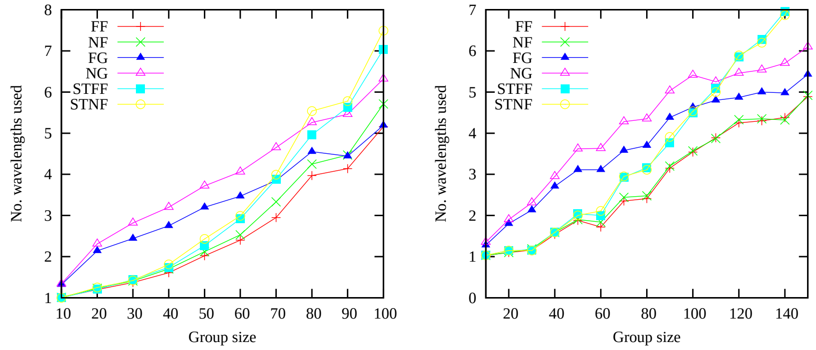
(a) N=200 (b) N=300

Fig. 8. Total Cost vs. Group size



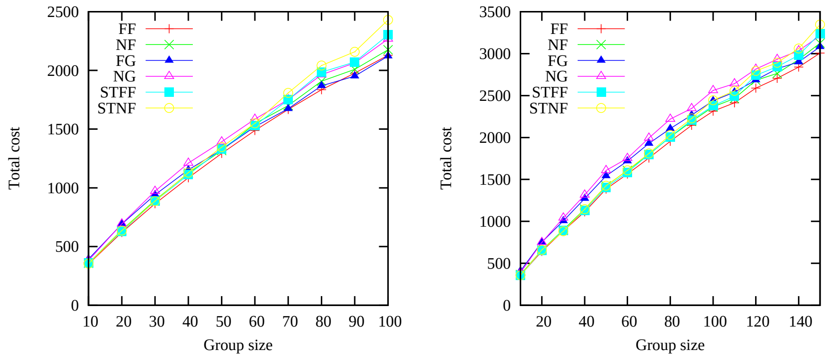
(a) N=200 (b) N=300

Fig. 9. Diameter vs. Group size



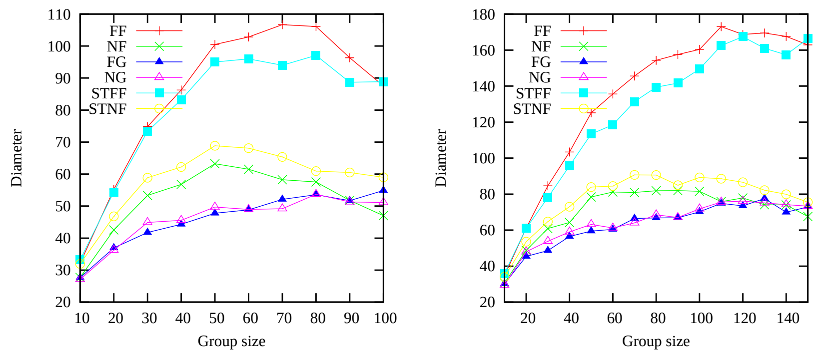
(a) N=200 (b) N=300

Fig. 10. No. Wavelengths vs. Group size



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

Fig. 11. Total Cost vs. Group size



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

Fig. 12. Diameter vs. Group size

6.5 Experimental result analysis

MDT versus MMDT

MDT and MMDT heuristics are just suitable in symmetric bidirectional networks in which they always result in optimal single wavelength but high cost and diameter. By taking advantage of multiple transmitters, MMDT heuristic also needs just one wavelength (Figures 7) and it is even better than MDT heuristic in term of cost (Figures 8) and diameter (Figures 9). 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 heuristics fail to get a solution.

Light-hierarchy based solutions versus light-tree based solutions

As seen in the experimental results above, light-hierarchy based solutions (FF and NF) outperform light-tree based solutions (FG and NG) 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 light-hierarchy based 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 maximum 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 maximum clique, and it will probably become the farthest one in the new maximum 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 maximum clique to diminish first, FF can reduce more cost than NF.

However, although FF chooses the longest trail first, the longest trail is not always necessarily chosen, due to the condition of arc-disjoint paths. Thus, the difference of the performance metrics (except the diameter) analysed above may not be considerable.

DSPT-based solution versus DAST-based solution

As shown in Figure 10, 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 addressed 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. We proved that the problem

NP-hard, 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, we presented that, unlike the popular approaches which assume to work in symmetric networks, our algorithm can work well in arbitrary networks.

Our four heuristic algorithms are compared with the proposed ones in the literature, and the simulation results showed that our algorithms achieve low number of used wavelengths, low cost but quite high diameter. The experimental results also showed that, among our four heuristic algorithms, the solutions based on DSPT are better than those based on DAST in term of the number of required wavelengths, the cost and the diameter of the trails. Between the two DSPT-based 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 algorithms are 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 set of 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*, 18:1628–1638, 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. Biswanath Mukherjee. *Optical WDM Networks*. Springer, 2006.
8. H. Takahashi and A. Matsuyama. An approximate solution for the Steiner problem in graphs. *Mathematica Japonica*, 24:573–577, 1980.

9. Yingyu Wan and Weifa Liang. On the minimum number of wavelengths in multicast trees in WDM networks. *Networks*, 45(1):42–48, 2005.
10. 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.
11. 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.
12. Yinzhu Zhou and Gee-Swee Poo. Optical multicast over wavelength-routed WDM networks: A survey. *Optical Switching and Networking*, 2(3):176 – 197, 2005.