Minimize Flow Interruptions during Reconfiguration of a set of Light-trees in All-optical WDM Network

Amanvon Ferdinand Atta[1†], Joël Christian Adépo[2††], Bernard Cousin[3†††], Souleymane Oumtanaga[4†]

[†]Laboratoire de Recherche en Informatique et Télécommunication, Institut National Félix Houphouët Boigny, Yamoussoukro, Côte d'Ivoire
[††]Unité de Recherche et d'Expertise Numérique, Université Virtuelle de Côte d'Ivoire, Abidjan, Côte d'Ivoire
[†††]Institut de Recherche en Informatique et Systèmes Aléatoires, Université de Rennes 1, Rennes, France

Summary
Reconfiguration of all-optical WDM networks is an important task performed by the network operators to re-optimize the allocation of network resources. Multicast applications such as video conference and telemedicine are increasingly in demand on the network. Thus, our study focuses on the reconfiguration of multicast connection paths. A multicast connection path is supposed to be a light-tree. The reconfiguration problem studied here is to find a sequence of configurations with minimum cost that allows to migrate an optical flow from a current configuration of a set of light-trees to a final configuration of a set of light-trees. To solve the problem, we propose a method that is based on a minimum cost feedback vertex set approach. The numerical results show that our method reduces flow interruptions more than a traditional minimum feedback vertex set approach.

Key words:
WDM network, Flow interruption, Light-tree reconfiguration, Multicast connection.

1. Introduction

1.1 Context

Wavelength Division Multiplexing (WDM) is basically a frequency division multiplexing in the optical frequency domain. On a single optical fiber, WDM creates multiple communication channels (called wavelength channel) at different wavelengths (corresponding to carrier frequencies) [1]. This technology is important to make it possible to meet the bandwidth requirements of applications such as videoconferencing, distance learning, IP-based TeleVision (IPTV) and telemedicine. A WDM network consists of a set of optical fibers connected by optical nodes [2] that use WDM as multiplexing technology.

The use of each of the above applications requires a multicast connection (i.e. the establishment of a communication structure between one source node and many destination nodes). In all-optical WDM network, a common algorithmic solution to establish multicast connection is a light-tree [3]. A light-tree can be defined as a tree-shaped path [4] formed by a set of wavelength channels. The tree-shaped path allows to transfer an optical flow (in an all-optically manner) from a source node to some destination nodes. Note that, each link of this tree-shaped path is called a wavelength channel. In addition, all wavelength channels of this tree-shaped path use the same wavelength. Due to the high popularity of multicast applications, many clients of a network operator require multicast connections. For each client connection request, the network operator configures a light-tree. In other words, the network operator may end up with many light-trees established on his network at a given time. For efficiency, the network operator must continually change the configuration of these light-trees: it is said that the operator performs reconfiguration of light-trees. Reconfiguration is unavoidable when an event such as a large number of connection requests occur on the network [5]. Reconfiguration responds to a need to optimize the allocation of network resources. However, if reconfiguration is performed improperly, then it may cause some interruptions of an optical flow.

1.2 Problem overview

Let M be a set of multicast connections such that each multicast connection is implemented by a current established light-tree which transport an optical flow. The set of currently established light-trees forms the current configuration. At the end of reconfiguration process, this set of currently established light-trees must be replaced by a set of final light-trees to be established. This set of final light-trees to be established forms the final configuration. The reconfiguration problem studied here is to find the sequence of configurations to migrate the optical flow from the current configuration to the final configuration with the minimum number of flow interruptions. Note that flow interruptions may compel the network operator to pay penalties to its clients [6].

This problem is tricky to solve because generally a final light-tree belonging to the final configuration

Manuscript received July 7, 2020
Manuscript revised July 20, 2020

requires a wavelength channel that is already used by a current light-tree belonging to the current configuration. This situation creates a deadlock state in the reconfiguration process.

For instance, let m_1,m_2 and m_3 be three multicast connections established on a WDM Network. We assume that only one wavelength channel can be established on each optical fiber. Fig. 1(a) contains the current light-tree (represented by the set of wavelength channels in solid red) of m_1, the current light-tree (represented by the set of wavelength channels in solid blue) of m_2 and the current light-tree (represented by the set of wavelength channels in solid green) of m_3. Also, Fig. 1(b) contains the final light-tree (represented by the set of wavelength channels in dotted red) of m_1, the final light-tree (represented by the set of wavelength channels in dotted blue) of m_2 and the final light-tree (represented by the set of wavelength channels in dotted green) of m_3.

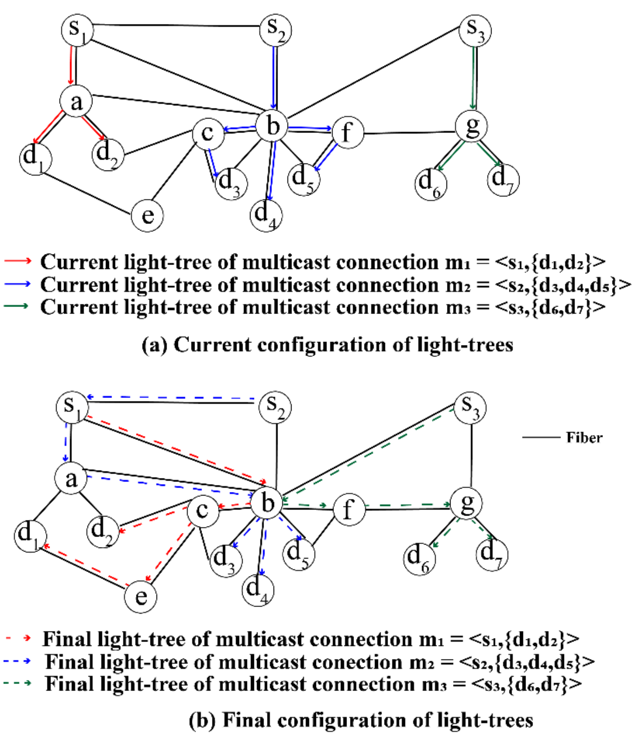

Fig. 1 Illustrative example of reconfiguration problem

We assume that all wavelength channels of both (current and final) configurations use the same wavelength. Indeed, the final light-tree of m_2 requires the wavelength channel $s_1 \rightarrow a$ that is already used by the current light-tree of m_1. Also, the final light-tree of m_1 requires the wavelength channel $b \rightarrow c$ that is already used by the current light-tree of m_2. This situation creates a deadlock state in the reconfiguration problem. A deadlock state cannot be overcome without some flow interruptions toward some destination nodes of some multicast connections.

1.3 Contributions and paper organization

To minimize flow interruptions during reconfiguration process, our study proposed a two-fold contribution: a) a vertex-weighted dependency graph to model deadlock state between light-trees and b) an algorithm based of a Minimum Cost Feedback Vertex Set Approach (MCFVSA). This approach overcomes deadlock state in reconfiguration process by interrupting flow toward the destination nodes of the multicast connections that composed the feedback vertex set [7] having the minimum cost.

After an introduction to the problem tackled in this paper, the reader can follow the rest of our paper in the following order. Section 2 presents the specification of our problem. In Section 3, we present related previous methods. Section 4 presents our contributions. Section 5 introduces a set of numerical experiments that evaluate the performance of our method. Section 6 summarizes our contributions and the performance of our method.

2. Problem specification

2.1 Model and assumptions

The WDM network is modeled as an undirected graph $G = (V, E)$ where V is the set of optical nodes and E is the set of fiber links.

Let $M = \{m_1, m_2, \dots, m_n\}$ be a set of established multicast connections in the WDM network where $n = |M|$. Each multicast connection $m_k \subseteq M$ is also denoted by $m_k = < s_k, D_k >$ where s_k is the source of the multicast connection m_k and $D_k = \{d_k^1, \dots, d_k^{i_k}\}$ is the set of destination nodes of m_k. The optical flow of the multicast connection m_k is initially carried by the current light-tree $T_k^0 = (V_k^0, E_k^0, \lambda_k^0)$. Note that $V_k^0 \subseteq V$, $s_k \subseteq V_k^0, D_k \subseteq V_k^0 \setminus \{s_k\}, E_k^0 \subseteq E$ and λ_k^0 is the wavelength used to create the wavelength channels of T_k^0 on the elements of E_k^0. At a given time, the network operator decides to do reconfiguration task. Therefore, for each multicast connection m_k of M, he computes a final light-tree $T_k^f = (V_k^f, E_k^f, \lambda_k^f)$ that will be used to carry the optical flow of m_k at the end of reconfiguration process. Note that $V_k^f \subseteq V$, $s_k \subseteq V_k^f, D_k \subseteq V_k^f \setminus \{s_k\}$, $E_k^f \subseteq E$ and λ_k^f is the wavelength used to create the wavelength channels of T_k^f on the elements of E_k^f.

The following assumptions are made:

(i) Each optical node has the multicast capability.
(ii) Each optical node does not have wavelength conversion capability.
(iii) Spare resources [8] is not available during reconfiguration process. This means that each connection cannot use a temporary light-tree.
(iv) According to the Service Level Agreements (SLAs), all multicast connections have the same management priority from the operator's point of view.

2.2 Problem formulation

The problem is formulated as follows:
(i) **Given**: A set of multicast connection $M = \{m_1, \ldots, m_n\}$, the current configuration of light-trees $C_0 = \{T_1^0, \ldots, T_n^0\}$ for M and the final configuration of light-trees $C_f = \{T_1^f, \ldots, T_n^f\}$ for M where T_k^0 and T_k^f is respectively the current light-tree of multicast connection m_k and the final light-tree of m_k.
(ii) **Goal**: Find the sequence of configurations denoted by $C = < C_0, C_1, \ldots, C_f >$ where $C_i = \{T_1^i, \ldots, T_n^i\}$ is the configuration of light-tree set used by the set of multicast connections M at step i of the reconfiguration process. Specifically, T_k^i is the light-tree used to carry the optical flow of the multicast connection m_k at step i of reconfiguration process. C must involve a minimum of the reconfiguration cost.

The reconfiguration cost considered here is the total number of flow interruptions produced by the reconfiguration process. This cost is denoted by $\mathrm{FLOW\_INT}(C)$. Let $FLOW\_\mathrm{INT}(C_i, m_k)$ be the number of destinations nodes of multicast connection m_k that does not receive the optical flow in the configuration C_i. $FLOW\_INT(C_i, m_k)$ is computed by Eq.(1):

$$\mathrm{FLOW\_INT}(C_i, m_k) = \sum_{\mathrm{d} \in \mathrm{D}_k} INT^{C_i}(d) \quad (1)$$

Where:

$$INT^{C_i}(d) = \begin{cases} 1 \text{ if the light-tree } T_k^i \in C_i \text{ does not span } d \\ 0, otherwise \end{cases}$$

Let $FLOW\_INT(C_i)$ be the number of flow interruptions caused by the configuration C_i. In others words, $FLOW\_INT(C_i)$ is the total number of flow interruptions caused by the set of multicast connections M at step i. $FLOW\_INT(C_i)$ is computed by Eq.(2) :

$$FLOW\_INT(C_i) = \sum_{m_k \in M} FLOW\_INT(C_i, m_k) \quad (2)$$

From Eq.(1) and Eq.(2), it is deduced that $\mathrm{FLOW\_INT}(C)$ is computed by Eq.(3):

$$\mathrm{FLOW\_INT}(C) = \sum_{C_i \in C} FLOW\_INT(C_i) \quad (3)$$

3. Related works

Reconfiguration is a task that concerns both multicast and unicast connections. A multicast connection is represented by a light-tree and a unicast connection is represented by a lightpath [9]. Hence, we summarize previous works related to light-tree reconfiguration and then these related to lightpath reconfiguration.

3.1 Light-tree reconfiguration

Light-tree reconfiguration was studied in [10], [11]. The works in [10], [11] deal with the problem of migrating an optical flow from a current light-tree to a final light-tree without flow interruption. In [10], the authors proposed a method which consists in reconfiguring a pair of light-trees (a current light-tree, a final light-tree) by successively reconfiguring the pairs of branches extracted from the pair of light-trees. It should be noted that a branch of a tree is a path from the root node of the tree to one of its leaf nodes. Although it allows reconfiguration without flow interruption, this method generates a high reconfiguration time. To overcome this limitation, in [11] the proposed method reconfigures a set of branch pairs at each step of the reconfiguration process.

In [10] and [11], the authors studied the light-tree reconfiguration problem that concerns a single multicast connection. The single multicast connection is characterized by a pair of light-trees (a current light-tree, a final light-tree). However, the problem tackled in our study (see Section 2.2) concerns a set of multicast connections. Each multicast connection is characterized by a pair of light-trees (a current light-tree, a final light-tree). One solution to solve the problem stated in Section 2.2 may be to use the work of [11] to reconfigure the pairs of light-trees in parallel. However, the dependency of wavelength channels between some multicast connections illustrated in Section 1.2 shows that the solution mentioned above is not reliable.

3.2 Lightpath reconfiguration

Lightpath is a case of light-tree that contains a single destination node. Therefore, the problem of lightpath reconfiguration is a special case of the problem of light-tree reconfiguration. Hence, the problem of lightpath reconfiguration is also tricky to solve because some final lightpaths require wavelength channels that are already used by some current lightpaths. It is said that some final lightpaths depend on some current lightpaths. The dependencies are modeled by an unweighted dependency graph [12]. The unweighted dependency graph is a directed graph that is built as follows [13]:
(i) Each connection is a vertex of the unweighted dependency graph
(ii) If the final lightpath of connection m_1 needs some wavelength channels that are already used by the current lightpath of connection m_2, then we create an edge from vertex m_1 to vertex m_2.

An instance of the problem of lightpath reconfiguration can be solved without flow interruption if the unweighted dependency graph of this instance is acyclic. Note that each cycle shows a deadlock in the instance of the problem. One deadlock causes unavoidably flow interruptions.

The works in [13]–[16] tackle the problem of lightpath reconfiguration. Each algorithm proposed in these works is a Minimum Feedback Vertex Set based Algorithm (MFVSA). Given a directed graph $G(V,E)$, a Feedback Vertex Set (FVS) of G is a set of vertices $V' \subseteq V$ such that the graph induced by the vertices $V \setminus V'$ is acyclic. In other words, removal of all elements of V' breaks all the circuits of the unweighted dependency graph G. MFVSA can be summarized in the following steps [14] :

(i) **Step 1:** Build the unweighted dependency graph G

(ii) **Step 2**: Compute a Minimum Feedback Vertex Set denoted by MFVS. A minimum feedback vertex set is a feedback vertex set that has minimum cardinality [17]. Note that authors proposed a heuristic solution to computes MFVS, whereas in [13], [15] and [16] authors use an exact algorithm.

(iii) **Step 3**: For each connection belonging to the MFVS, its current lightpath is deleted.

(iv) **Step 4**: For each connection that does not belong to the MFVS, its final lightpath is established then its current lightpath is deleted.

(v) **Step 5**: For each connection belonging to the MFVS, its final lightpath is established.

Note that Step 2 ensures that at Step 3, there are a minimum number of flow interruptions. Also, Step 4 causes the same number of flow interruptions as Step 3. In addition, Step 5 restores the flows that were interrupted. Therefore, MFVSA is the best solution to solve the problem of ligthpath reconfiguration.

Each unicast connection (i.e. one source node and one destination node) belonging to MFVS causes one flow interruption at a step of MFVSA. However, each multicast connection concerns more than one destination node. Therefore, if a multicast connection belongs to MFVS then it causes more than one flow interruption at a step of MFVSA. So, the use of MFVS at Step 2 of MFVSA does not ensure that MFVSA always returns the best solution for the problem of light-tree reconfiguration tackled in our study.

Fig. 1(a) and Fig. 1(b) show that the final light-tree of the connection m_1 requires the wavelength channel $b \to c$ that is already used by the current light-tree of the connection m_2. Also, the final light-tree of the connection m_2 requires the wavelength channel $s_1 \to a$ that is already used by the current light-tree of the connection m_1 and the final light-tree of the connection m_3 requires wavelength channel $b \to f$ that is already used by the current light-tree of the connection m_2. These dependencies are modeled by the unweighted dependency graph illustrated by Fig. 2.

Fig. 2 Unweighted dependency graph of the instance of light-tree reconfiguration problem described in Fig. 1

According to Fig. 2, there are two Feedback Vertex Set (FVS): $FVS_1 = \{m_1\}$ and $FVS_2 = \{m_2\}$. If we assume that the Step 2 of MFVSA returns $FVS_2 = \{m_2\}$ as MFVS, then MFVSA caused three flow interruptions (i.e. flow interruption to d_3, d_4, d_5) at Step 3 and Step 4. Whereas two flow interruptions (i.e. flow interruption to d_1 and d_2) at Step 3 and Step 4 could have been caused by using FVS1 as MFVS in Step 2. So, if MFVSA selects $FVS_2 = \{m_2\}$ at Step 2 then MFVSA does not return the best solution for the instance of the problem of light-tree reconfiguration depicts by Fig. 1.

4. Proposed Algorithm

The previous section shows that MFVSA does not always return the best solution for the problem of light-tree reconfiguration studied here. Therefore, in this section, we present our algorithm. This algorithm is based on an approach called Minimum Cost Feedback Vertex Set (MCFVS). This approach ensures that a minimum number of destination nodes are deprived of flow. For this purpose, a vertex-weighted dependency graph is introduced to clearly show the dependencies between connections. In the following subsections, we describe first how to build the vertex-weighted dependency graph and then the proposed algorithm in detail.

4.1 Vertex weighted dependency graph

The vertex-weighted dependency graph is built as follows:
(i) its vertices and its edges are created as the unweighted dependency graph (see the first paragraph of section 3.2)
(ii) A weight is assigned to each vertex according to a weight function. Formally, the vertex-weighted dependency graph is denoted by $G_w(V,E,w)$ where V is the set of vertices, E is the set of edges and w is the weight function which is defined as follows:

$$w: V \to N^*$$
$$v \mapsto w(v)$$

where $w(v)$ is the cost of deletion of the light-tree of the connection corresponding to vertex v in the current configuration. Our algorithm must cause the minimum of flow interruptions to the destination nodes of the multicast connections. Therefore, intuitively, $w(v)$ denotes the number of destination nodes of connection corresponding to vertex v.

4.2 Light-tree Set Reconfiguration Algorithm (LSRA)

In this subsection, we propose the Light-tree Set Reconfiguration Algorithm (LSRA). This algorithm (i.e. Algorithm 1) takes as input the current configuration C_0, the final configuration C_f, and the set of multicast connections M. And it returns a sequence of configurations denoted by C that causes the minimum number of flow interruptions to the destination nodes. To do this, our method:

(i) Builds a vertex-weighted dependency graph (refer to line 1 of Algorithm 1) while MFVSA (see section 3.2) builds a vertex-unweighted dependency graph.

(ii) Temporarily deletes the light-tree of each connection belonging to a set of connections having the smallest cost denoted by MCFVS (refer to line 4 of Algorithm 1) while MFVSA deletes the light-tree from each connection belonging to a set of connections having the smallest cardinality denoted by MFVS.

(iii) Does not fully reconfigure the connections that do not belong to MCFVS before completing the reconfiguration of the connections belonging to MCFVS (refer from line 7 to line 18 of Algorithm 1) while MFVSA fully reconfigures the connections that do not belong to MFVS before completing the reconfiguration of the connections belonging to MFVS.

In summary, the first element of the sequence of reconfigurations returned by LSRA is C_0. The next element in this sequence is obtained by performing an operation (i.e. deletion) that concerns the elements of the minimum cost feedback vertex set (refer to line 4 of Algorithm 1). Note that the problem of computing a minimum cost feedback vertex set is NP-Hard [7]. Many heuristics such as Demetrescu and Finocchi's heuristic[18] exist to compute a minimum cost feedback vertex set. Then other elements of the sequence of configurations are obtained by performing an operation that concerns the independent vertices (refer to lines 10 and 15 of Algorithm 1). These independent vertices belong either to V^* ($V^* = V \setminus MCFVS$) or to MCFVS, where V is the set of vertices of the vertex-weighted dependency graph. Note that a vertex v of a directed graph G is an independent vertex if and only if the outdegree of v is equal to zero. In other words, an independent vertex represents a connection that does not depend on other connections at the current step of the reconfiguration process. If after processing the vertices belonging to V^*, some elements of MCFVS are not fully processed, then one configuration is added to the sequence of configurations to complete this sequence (refer to line 20 of Algorithm 1).

Algorithm 1 Light-tree Set Reconfiguration Algorithm (LSRA)

Input: $C_0 = \{T_1^0, \dots, T_n^0\}$, $C_f = \{T_1^f, \dots, T_n^f\}$, $\mathrm{M} = \{\mathrm{m}_1, \dots, \mathrm{m}_n\}$

Output: $\mathrm{C} = < \mathrm{C}_0, \dots, \mathrm{C}_\mathrm{f} >$

1: $G_w \leftarrow$ The vertex-weighted dependency graph in accordance with with C_0, C_f and M.

2: $\mathrm{MCFVS} \leftarrow$ Minimum cost feedback vertex set of G_w

3: $\mathrm{C} \leftarrow \mathrm{C}_0$. $\mathrm{curr\_conf} \leftarrow \mathrm{C}_0$

4: The current light-trees in C_0 that concern the elements of MCFVS are deleted in parallel from the old value of $\mathrm{curr\_conf}$ to obtain a new value of $\mathrm{curr\_conf}$. Then this new value is added to C.

5: $V^* \leftarrow V \setminus MCFVS$, where V is the set of vertices of G_w.

6: $\mathrm{V^*\_reconfigured} \leftarrow \emptyset$. $\mathrm{MFCVS\_reconfigured} \leftarrow \emptyset$.

7: **Repeat**

8: $V^*\mathrm{\_UnReconf\_graph} \leftarrow$ Sub-graph of G_w that is induced by $V^* \setminus \mathrm{V^*\_reconfigured}$.

9: $\mathrm{V^*\_Frees} \leftarrow$ The set of independent vertices of $V^*\mathrm{\_UnReconf\_graph}$

10: The elements of $\mathrm{V^*\_Frees}$ are reconfigured (i.e. current light-trees are replaced by final light-trees) in parallel to obtain a new value of $\mathrm{curr\_conf}$. Then this new value is added to C.

11: $\mathrm{V^*\_reconfigured} \leftarrow \mathrm{V^*\_reconfigured} \cup \mathrm{V^*\_Frees}$.

12: $\mathrm{V\_Unreconf\_graph} \leftarrow$ Sub-graph of G_w that is induced by $\mathrm{V} \setminus (\mathrm{V^*\_reconfigured} \cup \mathrm{MFCVS\_reconfigured})$

13: $\mathrm{MCFVS\_Frees} \leftarrow$ The set of independent vertices of $\mathrm{V\_Unreconf\_graph}$ such that each element of this set belongs to MCFVS

14: **If** $\mathrm{MCFVS\_Frees}$ is not empty **then**

15: The final light-trees in C_f that concern the elements of $\mathrm{MCFVS\_Frees}$ are established in parallel so they are added to $\mathrm{curr\_conf}$ to obtain a new value of $\mathrm{curr\_conf}$. Then this new value is added to C.

16: $\mathrm{MCFVS\_reconfigured} \leftarrow \mathrm{MCFVS\_reconfigured} \cup \mathrm{MCFVS\_Frees}$.

17: **End If**

18: **Until** $V^*\mathrm{\_reconfigured} = \mathrm{V^*}$

19: **If** $\mathrm{MCFVS\_reconfigured} \neq \mathrm{MCFVS}$ **then**

20: The final light-trees in C_f that concern the elements of $\mathrm{MCFVS} \setminus \mathrm{MCFVS\_reconfigured}$ are established in parallel so they are added to $\mathrm{curr\_conf}$ to obtain a new value of $\mathrm{curr\_conf}$. Then this new value is added to C.

21: **End If**

22: **Return** C

The set of vertices of the vertex-weighted dependency graph G_w denoted by V is such that:

$$\mathrm{V} = \mathrm{MCFVS} \cup V^* \quad (4)$$

The first element of the sequence of configurations C is C_0. Line 4, line 10 and Eq.(4) show that the last element of the sequence C does not contain any light-tree belonging to the current configuration C_0. Also, lines 10, 15, 20 and Eq.(4) show that the last configuration contains each light-tree of the final configuration. So, the last configuration is the final configuration. It results that C is a configuration sequence to migrate from the current configuration to the final configuration (Result 1). MCFVS is a feedback vertex set having the minimum cost. Therefore, Algorithm 1 causes the minimum number of flow interruptions to destination nodes during the reconfiguration process (Result 2). Result 1 and Result 2 proves the partial correctness of Algorithm 1 (Result 3). V^*_reconfigured is initially empty (refer to line 6). In addition, at each iteration of the loop "Repeat", the elements of V^*_Frees are added to V^*_reconfigured (refer to line 11). Note that V^*_Frees is already a subset of V^* (refer to lines 8 and 9). Thus, at a certain iteration of the loop "Repeat", V^*_reconfigured becomes equal to V^*: the loop "Repeat" (refer from line 7 to line 18) stops. Therefore Algorithm 1 also stops (Result 4). Result 3 and result 4 prove the total correctness of Algorithm 1.

5. Experimental Consideration

Experiments have been conducted to confirm the effectiveness of our algorithm. The Python language is used to conduct these experiments on a laptop computer equipped with an Intel Core i5-9300H processor at 2.40 GHz, 8 GB RAM and a Windows 10 operating system. The metric evaluated in the experiments, the settings and the results of these experiments are presented in the following subsections.

5.1 Experimental metric

The metric used here to compare our algorithm (i.e. LSRA) with Minimum Feedback Vertex Set based Algorithm (MFVSA) is the total number of flow interruptions to the destination nodes of the set of multicast connections. Note that this metric is specified by Eq.(3) of Section 2.2.

5.2 Experimental Setup

To conduct the experiments, we used randomly generated networks. This ensures that the experiments results are independent of the characteristics of any practical network topology. We run LSRA and MFVSA on a randomly generated 200-node test network.

To generate the random graph we adopt the well-known Waxman approach [19] that is applied to WDM network and explained in [20]. In this approach, the vertices are placed randomly in a rectangular coordinate grid by generating uniformly distributed values for their *x* and *y* coordinates. An edge between two nodes *u* and *v* is added by using the probability function:

$$\mathrm{F} = \lambda \exp\left(-\frac{\mathrm{d(u,v)}}{\gamma\delta}\right) \quad (5),$$

where $\mathrm{d(u,v)}$ is the Euclidian distance between the nodes that form the endpoints of edge $(\mathrm{u,v})$, δ is the maximum Euclidian distance between any two nodes, $0 < \lambda, \gamma \leq 1$. Higher values of λ produce graphs with higher edge densities, while small values of γ increase the density of short edges relative to higher ones. In the experiments, λ and γ both are set respectively to 0.7 and 0.9.

WDM network represented by the random graph was used to generate instances of the problem of light-tree reconfiguration. We consider two experiment parameters: the number of multicast connections $|M| = n$ ($n_1 = 5$, $n_2 = 15$, $n_3 = 25$) and the range of the number of destination nodes of any multicast connection: ΔD ($\Delta\mathrm{D}_1 =[2;10]$, $\Delta\mathrm{D}_2 = [11;20]$, $\Delta\mathrm{D}_3 = [21:30]$). In other words, an experiment consists to take a value of the pair of parameters $(n_i, \Delta D_i)$, to generate 100 instances of the problem of light-tree reconfiguration, to run the algorithms on each of these instances and to assess the average of the experiment metric. Each instance of the problem is generated as follows:

(i) n_i multicast connections are generated: n_i source nodes are uniformly picked in [1;200] and for each multicast connection the number of destinations is picked in ΔD_i;

(ii) The current configuration is built: For each multicast connection, the light-tree is obtained by using Dijkstra algorithm [21] and used one wavelength.

(iii) The final configuration is built: For each multicast connection, the light-tree is obtained by using Prim algorithm [22] and used one wavelength such that each multicast connection depends on two other multicast connections. These two configurations (i.e. current and final) are taken as input by our algorithm (i.e. LSRA) and MFVSA.

5.3 Experimental Results

In this subsection, the performance of our proposed LSRA (i.e. our algorithm) and the performance of the previous approach MFVSA (see Section 3.2) are compared. We remind that, for each value of the range for the number of destination nodes ΔD ($\Delta\mathrm{D}_1 = [2;10]$, $\Delta\mathrm{D}_2 = [11;20]$, $\Delta\mathrm{D}_3 = [21;30]$) of any multicast connection, the number

of multicast connections $|M|$ are successively set at 5, 15 and 25. To gather representative results, for each value of $|M|$, 100 instances of reconfiguration problem are generated, then the average of the number of flow interruptions is computed. Thus, Fig. 2, Fig. and Fig. show the average of the number of flow interruptions according to the different values of the number of multicast connections $|M|$ (5, 15 and 25) when ΔD takes as value respectively [2;10], [11;20] and [21;30].
From Fig. 2 to Fig. , the followings observations are made:
(i) Regardless of the value of ΔD considered, our algorithm (LSRA) produces fewer flow interruptions than MFVSA. The reason is that, LSRA deletes temporarily the light-trees of a feedback vertex set having the smallest cost whereas MFVSA deletes temporarily a feedback vertex set having the smallest cardinality. Indeed, a feedback vertex set having the smallest cardinality has a cost that is greater than or equal to a feedback vertex set having the smallest cost.
(ii) For each value of ΔD, the number of flow interruptions increases as the number of connections $|M|$ increases. There are two reasons for this observation: a) More connections imply that the minimum value of the cost of a FVS is high. Indeed, more connections means that there are more dependencies and therefore more cycles in the dependency graph. b) Eq.(2) shows that the number of flow interruptions for a configuration is the sum of the number of flow interruptions for each connection. Therefore, more connections logically imply more flow interruptions.
(iii) For a given number of multicast connections, the number of flow interruptions increases as the number of destination nodes increases. This is consistent with Eq.(1), Eq.(2) and Eq.(3), which are used to determine the number of flow interruptions.

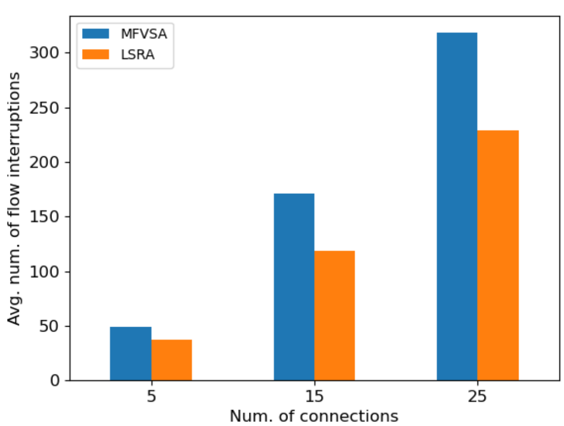

Fig. 2 Comparison of the average number of flow interruptions when the range of the number of destination nodes is [2;10]

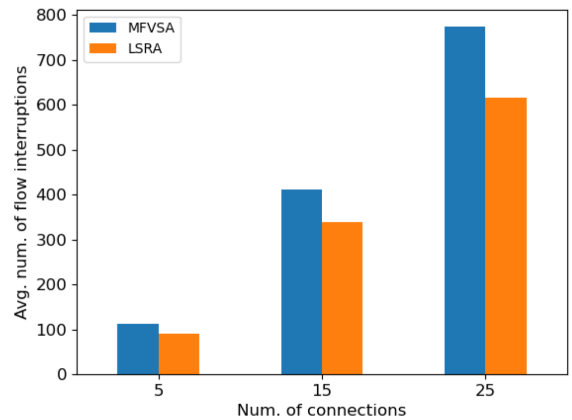

Fig. 3 Comparison of the average number of flow interruptions when the range of the number of destination nodes is [11;20]

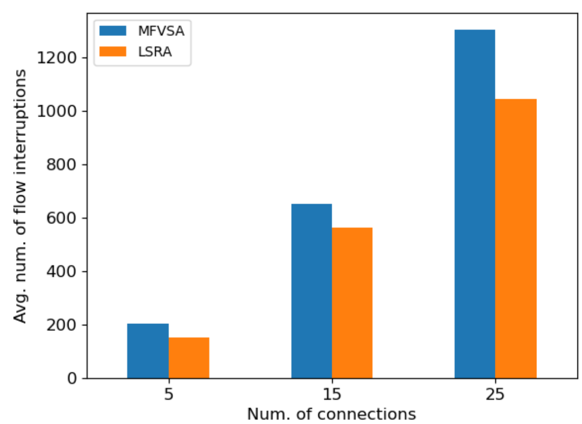

Fig. 4 Comparison of the average number of flow interruptions when the range of the number of destination nodes is [21;30]

6. Conclusion

Reconfiguration is a task that improves resource utilization in WDM network. The main point of this study is how to migrate from a current configuration of a light-tree set to a new one with a minimum number of flow interruptions. First, we have introduced a vertex-weighted dependency graph to model all dependencies. Next, based on a Minimum Cost Feedback Vertex Set Approach (MCFVSA), we have proposed an algorithm to find the configuration sequence that minimizes flow interruptions to destination nodes.
Finally, the effectiveness of the proposed algorithm has been confirmed experimentally.

Note that a semi-light-tree [23] is another type of tree-shaped path for multicast connection that uses more than one wavelength. Therefore, future works can tackle the reconfiguration problem that consists to migrate an optical flow from a semi-light-tree to a new one.

References

[1] R. Ramaswami and G. Sasaki, 'Multiwavelength optical networks with limited wavelength conversion', IEEE/ACM Trans. Netw., vol. 6, no. 6, pp. 744–754, Dec. 1998, doi: 10.1109/90.748086.

[2] Q. Wu, X. Zhou, J. Wang, Z. Yin, and L. Lin, 'Multicast routing and wavelength assignment with delay constraint in WDM networks with sparse wavelength conversions', Photonic Netw. Commun., vol. 19, no. 2, pp. 144–154, Apr. 2010, doi: 10.1007/s11107-009-0219-5.

[3] S. Barat, B. N. Keshri, and T. De, 'A Cost Function Based Multi-objective Multicast Communication over WDM Optical Fiber Mesh Network', in Advances in Computer, Communication and Control, vol. 41, U. Biswas, A. Banerjee, S. Pal, A. Biswas, D. Sarkar, and S. Haldar, Eds. Singapore: Springer Singapore, 2019, pp. 75–85.

[4] Hwa-Chun Lin and Sheng-Wei Wang, 'Splitter placement in all-optical WDM networks', in GLOBECOM '05. IEEE Global Telecommunications Conference, 2005., Nov. 2005, vol. 1, p. 5 pp.-, doi: 10.1109/GLOCOM.2005.1577639.

[5] J. Wu, 'A survey of WDM network reconfiguration: Strategies and triggering methods', Comput. Netw., vol. 55, no. 11, pp. 2622–2645, Aug. 2011, doi: 10.1016/j.comnet.2011.05.012.

[6] M. E. M. Saad and Zhi-Quan Luo, 'Reconfiguration with no service disruption in multifiber WDM networks based on Lagrangean decomposition', in IEEE International Conference on Communications, 2003. ICC '03., Anchorage, AK, USA, 2003, vol. 2, pp. 1509–1513, doi: 10.1109/ICC.2003.1204643.

[7] P. Festa, P. M. Pardalos, and M. G. C. Resende, 'Feedback Set Problems', in Handbook of Combinatorial Optimization, D.-Z. Du and P. M. Pardalos, Eds. Boston, MA: Springer US, 1999, pp. 209–258.

[8] A. Kadohata, A. Hirano, F. Inuzuka, A. Watanabe, and O. Ishida, 'Wavelength Path Reconfiguration Design in Transparent Optical WDM Networks', J. Opt. Commun. Netw., vol. 5, no. 7, p. 751, Jul. 2013, doi: 10.1364/JOCN.5.000751.

[9] T. Shuai and W. Ai, 'New algorithms for multicast routing and wavelength assignment in multi-hop optical WDM networks', Photonic Netw. Commun., vol. 23, no. 1, pp. 53–59, Feb. 2012, doi: 10.1007/s11107-011-0335-x.

[10] B. Cousin, J. C. Adépo, S. Oumtanaga, and M. Babri, 'Tree reconfiguration without lightpath interruption in WDM optical networks', Int. J. Internet Protoc. Technol., vol. 7, no. 2, pp. 85–95, 2012.

[11] J. C. Adépo, B. Aka, and M. Babri, 'Tree Reconfiguration with Network Resources Constraint', Int. J. Comput. Sci. Telecommun., vol. 7, no. 1, pp. 1–4, Jan. 2016.

[12] F. Balmas, 'Displaying dependence graphs: a hierarchical approach', in Proceedings Eighth Working Conference on Reverse Engineering, Stuttgart, Germany, 2001, pp. 261–270, doi: 10.1109/WCRE.2001.957830.

[13] F. Solano, 'Slick Lightpath Reconfiguration Using Spare Resources', J. Opt. Commun. Netw., vol. 5, no. 9, p. 1021, Sep. 2013, doi: 10.1364/JOCN.5.001021.

[14] N. Jose and A. K. Somani, 'Connection rerouting/network reconfiguration', in Fourth International Workshop on Design of Reliable Communication Networks, 2003. (DRCN 2003). Proceedings., Banff, Alberta, Canada, 2003, pp. 23–30, doi: 10.1109/DRCN.2003.1275334.

[15] F. Solano, 'Analyzing two conflicting objectives of the wdm lightpath reconfiguration problem', in Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE, 2009, pp. 1–7.

[16] F. Solano and M. Pióro, 'Lightpath Reconfiguration in WDM Networks', J. Opt. Commun. Netw., vol. 2, no. 12, p. 1010, Dec. 2010, doi: 10.1364/JOCN.2.001010.

[17] Hen-Ming Lin and Jing-Yang Jou, 'Computing minimum feedback vertex sets by contraction operations and its applications on CAD', in Proceedings 1999 IEEE International Conference on Computer Design: VLSI in Computers and Processors (Cat. No.99CB37040), Austin, TX, USA, 1999, pp. 364–369, doi: 10.1109/ICCD.1999.808567.

[18] C. Demetrescu and I. Finocchi, 'Combinatorial algorithms for feedback problems in directed graphs', Inf. Process. Lett., vol. 86, no. 3, pp. 129–136, May 2003, doi: 10.1016/S0020-0190(02)00491-X.

[19] B. M. Waxman, 'Routing of multipoint connections', IEEE J. Sel. Areas Commun., vol. 6, no. 9, pp. 1617–1622, Dec. 1988, doi: 10.1109/49.12889.

[20] Xiao-Hua Jia, Ding-Zhu Du, Xiao-Dong Hu, Man-Kei Lee, and Jun Gu, 'Optimization of wavelength assignment for QoS multicast in WDM networks', IEEE Trans. Commun., vol. 49, no. 2, pp. 341–350, Feb. 2001, doi: 10.1109/26.905896.

[21] E. W. Dijkstra, 'A note on two problems in connexion with graphs', Numer. Math., vol. 1, no. 1, pp. 269–271, Dec. 1959, doi: 10.1007/BF01386390.

[22] R. C. Prim, 'Shortest Connection Networks And Some Generalizations', Bell Syst. Tech. J., vol. 36, no. 6, pp. 1389–1401, Nov. 1957, doi: 10.1002/j.1538-7305.1957.tb01515.x.

[23] Aijun Ding, Gee-Swee Poo, and Sun-Teck Tan, 'An expanded graph model for MCRWA problem in WDM networks', in 27th Annual IEEE Conference on Local Computer Networks, 2002. Proceedings. LCN 2002., Tampa, FL, USA, 2002, pp. 557–564, doi: 10.1109/LCN.2002.1181830.

Amanvon Ferdinand Atta received his master's degree in computer science engineering at Université Nangui Abrogoua (Côte d'Ivoire) in 2016. Currently, he is a PhD student in computer science at Institut National Polytechnique Felix Houphouët Boigny (INP-HB) in Côte d'Ivoire. His research interests include optical network and software defined networks.

Souleymane Oumtanaga received his PhD in Computer Science at Université Paul Sabatier (Toulouse, France) in 1995. He is full professor in computer science at INP-HB since 2007. He was the head of the Network Information Center of Côte d'Ivoire. His research interests include IPv6, wireless network and optical network.

Joël Christian Adépo received his PhD in Computer Science at Université Nangui Abrogoua (Côte d'Ivoire) in 2016. Currently, he is an assistant professor in Computer Science at Université virtuelle (Côte d'Ivoire). his research interests include routing reconfiguration.

Bernard Cousin received in 1987 his PhD degree in computer science from the University of Paris 6. He is full professor of Computer Science at the University of Rennes 1, France. He is the co-author of a net-work technology book: "IPV6"(Fourth edition, O'Reilly, 2006) and has co-authored a few IETF drafts in the areas of Explicit Multicasting and Secure DNS. His research interests include all-optical networks, network security and mobile networks.