

Alma Mater Studiorum Università di Bologna  
Archivio istituzionale della ricerca

Load balancing in fixed-routing optical networks with weighted ordering heuristics

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

*Published Version:*

Bonani L.H., Queiroz J.C.F., Abbade M.L.F., Callegati F. (2019). Load balancing in fixed-routing optical networks with weighted ordering heuristics. JOURNAL OF OPTICAL COMMUNICATIONS AND NETWORKING, 11(3), 26-38 [10.1364/JOCN.11.000026].

*Availability:*

This version is available at: <https://hdl.handle.net/11585/785322> since: 2020-12-23

*Published:*

DOI: <http://doi.org/10.1364/JOCN.11.000026>

*Terms of use:*

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (<https://cris.unibo.it/>).  
When citing, please refer to the published version.

(Article begins on next page)

This is the final peer-reviewed accepted manuscript of:

**L. H. Bonani, J. C. F. Queiroz, M. L. F. Abbade, and F. Callegati, "Load Balancing in Fixed-Routing Optical Networks with Weighted Ordering Heuristics," J. Opt. Commun. Netw. 11, 26-38 (2019).**

The final published version is available online at: <http://dx.doi.org/10.1364/JOCN.11.000026>

#### Rights / License:

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

*This item was downloaded from IRIS Università di Bologna (<https://cris.unibo.it/>)*

***When citing, please refer to the published version.***

# Load Balancing in Fixed-Routing Optical Networks with Weighted Ordering Heuristics

L. H. Bonani, *Member, IEEE*, J. C. F. Queiroz, M. L. F. Abbade, *Member, IEEE* and F. Callegati, *Senior Member, IEEE*.

**Abstract**—In this paper we show that the adoption of a simple weighted ordering heuristic strategy to find fixed shortest path routes in a topology can improve the load-balancing and, consequently, the network performance. We compared our fixed routing strategies against  $k$ -shortest paths ( $k$ -SP) fixed-alternate routing strategy, for three different mesh topologies considering fixed grid networks (FGN) and elastic optical networks (EON). The results show that our simple fixed routing strategies can improve the network performance for an optical network system, even compared with the fixed-alternate routing, and always using the shortest path. Results also show that such easy strategies act mainly in longer routes, increasing their probability of use, and balancing the link load distribution.

**Index Terms**—Routing, Optical Networks.

## I. INTRODUCTION

CORE communication networks are based on wavelength division multiplexing (WDM) technology, usually implemented over a transparent optical network (TON) infrastructure. Presently TONs rely on a WDM fixed grid network (FGN) where channels are distributed over homogeneously spaced optical carriers. Communication is accomplished through a lightpath constituted by a channel allocation over a set of optical links between the edge nodes. This FGN approach will possibly evolve to an elastic optical network (EON) paradigm where a variable bandwidth channel, instead of a fixed channel, is assigned over the links connecting the edge nodes [1]. The variable bandwidth channel comprises several contiguous frequency slot units (FSU) and it is referred as a superchannel. Individual superchannels support different services, conveying signals with specific bit rates, modulation formats, and optical carriers/subcarriers. In order to setup the lightpath (either fixed or variable bandwidth), a control plane chooses the network resource, that can be a single wavelength for FGNs or a spectrum slice for EONs.

The process of setting up the lightpath is generally treated by solving separately two subproblems: routing assignment and resources assignment. Thus, the set of these two subproblems is known as routing and wavelength assignment (RWA) problem in FGNs [2] and routing and spectrum assignment (RSA) problem in EONs [3][4]. These problems have been widely studied in the past. Nonetheless, being the general

problem too complex for a comprehensive optimal solution, effective heuristics may still provide useful insights and efficient solution to specific subproblems.

This is the scope of this paper, where we propose a new approach to perform fixed routing for FGNs and EONs, improving the load balancing and keeping the route hop distance unchanged (in comparison to basic route definition). These characteristics are achieved by means of using a simple weighted ordering heuristic to perform the routes definition. The basic idea is the following. Initially some heuristic is applied to determine a sequence of ordering pairs of nodes. These pairs of nodes designate the source and destination nodes that may be used in a given topology. The shortest paths is then computed for this sequence of node pairs. However, whenever a link is used in a given route, its weight is increased. This leads to a penalization in the calculation of further shortest paths. We refer to this strategy as a weighted ordering heuristic (WOH). It presents some significant characteristics:

- It is simple to implement. It requires no extra hardware and, as a fixed routing technique, software runs only during the network booting;
- The computation complexity is relatively low. It only requires the achievement of an ordering scheme for the pairs of nodes and a new round of weighted route calculations (using a given shortest path routing algorithm like Dijkstra's);
- It enhances the network performance, as shown in Section V. For instance, for a FGN and blocking probability of 1%, the proposed strategy may improve the network load between 4% and 26%, depending on the topology;
- It keeps the routes in their minimum lengths, avoiding the longer routes introduced, for instance, by  $k$ -SP fixed-alternate routing technique.

To the best of our knowledge, this is the first time that such simple and useful approach is reported in the literature. The work is organized as follows. In Section II a brief summary of significant related works is provided. Then, in Section III we present some definitions about ordering of node pairs and the strategy of getting routes with WOH to obtain the fixed routes with load balancing for a given topology. In Section IV-A we present a network load normalization method for FGN and EON paradigms. These are practical definitions that, to the best of our knowledge, are not well-defined in literature, and that may be useful for other works as well. Section IV-B presents the performance metrics adopted for this work. In Section V the numerical results are presented and discussed. Specifically,

L. H. Bonani and J. C. F. Queiroz are with Center for Engineering, Modeling and Applied Social Sciences, Universidade Federal do ABC, Santo André, Brazil (e-mail: {luiz.bonani, joel.queiroz}@ufabc.edu.br)

M. L. F. Abbade is with Universidade Estadual Paulista (UNESP), São João da Boa Vista, Brazil (marcelo.abbade@unesp.br)

F. Callegati is with University of Bologna, Cesena, Italy (e-mail: franco.callegati@unibo.it).

Section V-A presents the adopted network scenarios for the tests and Section V-B presents the performance analysis, including discussions about statistics of topologies. Finally, Section VI presents the conclusions supported by our results.

## II. ROUTING AND RELATED WORKS

As outlined above, RWA and RSA require routing and resources assignment. For the routing subproblem, there are three strategies that impact on the network operation and on the chosen route for a given lightpath: (i) fixed routing, (ii) fixed-alternate routing and (iii) adaptive routing, which are deeply studied in [5] and [6]. The resource assignment subproblem presents different complexities regarding FGNs or EONs.

For FGNs based on WDM technology, an available wavelength along a whole route (set of links) must be found using heuristic RWA algorithms [1]. The complexity of these algorithms depends on whether wavelength conversion is used or not. If wavelength conversion is possible, the wavelength may change from one link to another and RWAs are relatively simple. However, this operation demands the utilization of wavelength converters, which are complex and increase the network cost. For this reason, practical FGNs do not deploy wavelength conversion and the same wavelength must be available along the whole route. This scenario poses the so-called *wavelength continuity* constraint, which enhances the complexity of RWAs. From now on, in compliance with realistic WDM implementations, in this paper we always assume that wavelength conversion is not available.

In the context of EONs, the wavelength continuity constraint is referred as the *spectrum continuity* constraint and must also be obeyed. A spectral slice is a set of contiguous FSUs that support a given line rate with an appropriate modulation format. Thus, all FSUs in a slice must be contiguously free through the whole route. This additional condition, called the *spectrum contiguity*, must also be treated by RSAs, enhancing their complexity.

From the logical point of view, the most significant performance measure is the connection blocking probability resulting from resource unavailability when dynamic requests occur. A connection request is blocked if there are not enough available wavelengths or spectrum slices due to a congested route link. This problem is critical under high load conditions and it can be treated using a set of multiple routes between each source-destination pairs of nodes, which characterizes the fixed-alternate routing strategy [7]–[11]. For instance, [7] proposes a reduction of blocking probability by means of an fixed-alternate routing algorithm which uses a pool of routing paths formed by link-disjoint between sources and destinations. Authors of [8] shows an improvement reached with the use of an fixed-alternate routing method in which is made a kind of reservation. This strategy allows alternative routes for connections with more hops. Working on routing and wavelength assignment, [9] proposes a scheme based on priority order, firstly on the type of path, and then on the traffic amount. This technique reduces the level of blocking probability that happens due to the wavelength continuity

constraint. In [10], authors rearrange the routes according to a defined cost, achieving lower network blocking probabilities. Also, authors of [11] addressed the problem of dynamic routing of anycast demands in EONs, reporting detailed results to show the trade-offs between fixed routing and fixed-alternate routing approaches, showing that the last one tends to be more interesting to EONs. Besides the advantages, fixed-alternate routing is more complex to manage and it tends to degrade the network performance in higher loads.

When the routing tables change dynamically as function of current network state, we have the adaptive routing techniques [12]–[14]. In such a context, authors in [12] propose an algorithm for fixed-alternate routing, which writes a routing table according to the load distribution and to the location of each source-destination pair. They show an enhanced performance measured by blocking probability. Moreover, [13] works on an optimized analytic model for performance measurement of shortest path routing in WDM networks, with and without wavelength conversion. The authors in [14] propose a routing scheme that uses the benefit of distance-adaptive modulation and an adaptive bit rate capability inherited by EON, to improve spectrum utilization. All these fixed-alternate routing and adaptive routing strategies demands many information from network state, as well as complex software, and also dynamic actions in the adaptive routing case, being very expensive. Furthermore, these techniques tend to increase the average number of hops due to utilization of longer routes.

The simpler routing strategy is the fixed routing [15]–[16], where a single fixed route between each source and destination node pair is determined *a priori*. This fixed route, obtained by a shortest path algorithm like Dijkstra's [17], is configured in the routing devices and it does not change during network operation. Therefore, when a connection request must be set up between any pair of nodes, the route is at hand for finding a possible network resource. If a free resource (wavelength or spectrum slice) is available throughout the given route, the connection setup is complete. On the contrary, if there are no free network resources, the connection is blocked. Many studies have addressed ways to get fixed routing [18]–[25]. For instance, in [18] it is shown that longer routes have a higher blocking penalty due to certain congested links when using a fixed routing strategy. In [19] authors evaluate the restricted routing technique (RRT) [20], that was evaluated in networks with complete wavelength conversion capacity, working to decrease the congestion in some highly used links. However, to achieve this objective, RRT operates with routes that do not follow the shortest path metric. Other strategy [21] is based on M-combinations of shortest paths (MCSP), aiming at finding new shortest paths that balance the utilization of all network links. In spite of its performance advantages, this is an exhaustive method whose complexity increases with the topology size. In [22], the authors propose a load-balanced fixed routing (LBFR) [23] scheme that avoids network congestion and retains the operational simplicity of fixed routing. The algorithm is based on the assumption that future traffic demands can be predicted, depending on the knowledge of applied traffic and on the amount of available resources. Authors of [24] propose a load balancing based on

both routing and wavelength assignment. In this strategy, the most used and least used wavelength algorithms are combined depending on the topology features and on the characteristics of network traffic. However, there is no guarantee that only shortest path routes are used. In [25], authors approach the problem of multicast routing. Similarly to an adaptive routing strategy, their load balancing technique considers the current load distribution of the network for the routing definition of the upcoming requests. So, they achieve a balanced distribution of load with the least congested links being utilized for the routing of the upcoming requests, while the highly congested ones are avoided.

Therefore, from the achievements concerning routing techniques, we expect that there are still opportunities to a simple and efficient fixed routing strategy using weighted ordering heuristics.

### III. THE WEIGHTED ORDERING HEURISTIC (WOH)

Some basic definitions are needed to present this work. So,  $G(N, E)$  is the graph of a network topology, with  $N = \{n_i \mid 1 \leq i \leq n\}$  being the set of nodes and  $E = \{e_i \mid 1 \leq i \leq e\}$  being the set of edges (links). Each link  $e_i$  can also be represented by its input node,  $n_{in}^{e_i}$ , and its output node  $n_{out}^{e_i}$ .

The topology connectivity matrix  $\mathbf{M}$  is an  $n$ -squared matrix. The row and column indexes relate to the source ( $s$ ) and destination ( $d$ ) nodes, respectively. The elements ( $s, d$ ) of  $\mathbf{M}$  are equal to 1 if there is a link between nodes  $s$  and  $d$ , and equal to 0 otherwise. Being the application of this work tailored to optical networks, we assume a directed  $G$  graph, in which the link direction matters and the link from  $s$  to  $d$  is different from the link from  $d$  to  $s$ . This is in compliance with the simplex configuration of practical optical fiber links. Without loss of generality for the presentation, we assume links have all the same weight equal to 1. Then a *route*, that is a path connecting any two pairs of nodes in  $G$ , can be represented by an array of nodes starting from  $s$  and arriving to  $d$ , so  $\mathbf{r}(s, d) = [s \dots d]$ . The route length, computed by function  $Length()$ , can be measured by its number of links (or number of hops), which is the number of node elements in  $\mathbf{r}(s, d)$  minus 1:  $Length(\mathbf{r}(s, d)) = |\mathbf{r}(s, d)| - 1$ .

Thus, any topology has at least two main routes between any pair of nodes  $s$  and  $d$ . The first one is the forward route from  $s$  to  $d$ ; the second one is the backward one, from  $d$  to  $s$ . It is assumed that forward and backward routes between nodes  $s$  and  $d$  differ only in the flow direction, passing through the same nodes. So, if we have the forward route, we can get the backward one using a  $Flip()$  function, that flips the route array  $\mathbf{r}(s, d)$  from the left to the right:  $Flip(\mathbf{r}(s, d)) = \mathbf{r}(d, s)$ .

Furthermore, if a route from  $s$  to  $d$  counts the minimum possible number of links (or nodes), it is a *shortest path*. For a given topology, there is a set of shortest paths,  $\mathbf{SP}$ , represented as an  $n$ -squared generic matrix, on which each element  $\mathbf{SP}(s, d)$  is the shortest path between nodes  $s$  and  $d$ . The total number of shortest path routes, including forward and backward ones, is given by  $n(n - 1)$ . However, if we are interested in the *unique pair of nodes* ( $s, d$ ), without considering the flow direction, the number is  $p_n = n(n - 1)/2$ .

In the remainder of this work it will be useful to have a formal way to address all the unique pair of nodes. Therefore we define the upper triangular square matrix

$$\mathbf{T} = \begin{bmatrix} 0 & 1 & 1 & \dots & 1 \\ 0 & 0 & 1 & \dots & 1 \\ 0 & 0 & 0 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}, \quad (1)$$

where a non-zero element  $\mathbf{T}(s, d)$  represents the unique pair of nodes ( $s, d$ ) that can be connected through the topology. Moreover, from any node  $s$  to any node  $d$ , the shortest path  $\mathbf{SP}(s, d)$  can be calculated using, for instance, the Dijkstra's algorithm. Generally speaking, all  $\mathbf{SP}$ s are calculated as isolated problems and a given route does not depend on those already calculated and/or on the order they are calculated. The proposal of this work is based on two simple concepts:

- *ordering* of unique pair of nodes in a  $p_n \times 2$  bi-dimensional array (called Node Pair Order or **NpO**) according to some ordering logic;
- *ordered calculation* of the  $\mathbf{SP}$ s according to **NpO**;
- *update* of the link cost at any new  $\mathbf{SP}$  calculation, as a function of the previous calculation.

#### A. Node Pairs Order

In order to obtain the **NpO** we propose to proceed as follows:

- 1) calculate  $\mathbf{SP}(s, d)$  for any  $s$  and  $d$  in a given topology with conventional algorithms such as Dijkstra's, with symmetric forward and backward routes;
- 2) build a matrix **NpO** of size  $p_n \times 2$ , ordered according to some algorithm.

We consider all  $p_n$  unique pairs of nodes in matrix  $\mathbf{T}$  and define some simple ordering heuristics (OH) for **NpO**, as follows:

- RD: Random node pairs order;
- HoAS: Hop oriented with ascending and alternate source order;
- HoAD: Hop oriented with ascending and alternate destination order;
- HoRD: Hop oriented with ascending and random order.

In order to have a general algorithm to achieve **NpOs** with any OH, firstly we use Algorithm 1 to construct a dynamical list of elements in  $\mathbf{T}$ , called list of unique pairs of nodes,  $T_{list}$ . With  $T_{list}$ , we can access or remove any specific node pair element represented by an  $[s \ d]$  array. The number of elements in  $T_{list}$ , whose maximum value is  $p_n$ , can be achieved with a  $Size()$  function. Since  $\mathbf{T}$  is characterized by having  $n - 1$  non-zero elements in the first row,  $n - 2$  ones in the second row, and so on,  $\mathbf{T}$  is not explicitly necessary to achieve  $T_{list}$ , as shown in Algorithm 1. Therefore, lines 2 and 3 of function  $GetListOfNodePairs()$  goes through the non-zero elements in  $\mathbf{T}$  (unique pairs of nodes), with just one restriction: if the input  $h$  is set to zero (line 4), all  $[s \ d]$  unique pair of nodes are added to  $T_{list}$  (line 5). Otherwise, if a different value is assigned to  $h$ , line 7 states that only the pairs of nodes which

**Algorithm 1** Getting the List of Node Pairs,  $T_{\text{list}}$ 


---

```

1: Function GetListOfNodePairs( $\mathbf{SP}, n, h$ )
2: for  $s \leftarrow 0, n-2$  do
3:   for  $d \leftarrow s+1, n-1$  do
4:     if  $h = 0$  then
5:        $T_{\text{list}} \leftarrow [s \ d]$ 
6:     else
7:       if  $\mathbf{SP}(s, d) = h$  then
8:          $T_{\text{list}} \leftarrow [s \ d]$ 
9:       end if
10:    end if
11:  end for
12: end for
13: return  $T_{\text{list}}$ 
14: end Function

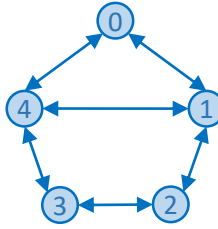
```

---

**SP Routes:**Unique Node Pair  $\rightarrow$  Shortest Path

(0,1)  $\rightarrow$  [0 1] (1)  
 (0,2)  $\rightarrow$  [0 1 2] (2)  
 (0,3)  $\rightarrow$  [0 4 3] (2)  
 (0,4)  $\rightarrow$  [0 4] (1)  
 (1,2)  $\rightarrow$  [1 2] (1)  
 (1,3)  $\rightarrow$  [1 2 3] (2)  
 (1,4)  $\rightarrow$  [1 4] (1)  
 (2,3)  $\rightarrow$  [2 3] (1)  
 (2,4)  $\rightarrow$  [2 1 4] (2)  
 (3,4)  $\rightarrow$  [3 4] (1)

## 5-node Topology



## Connectivity Matrix:

$$\mathbf{M} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

## Unique Pairs Matrix:

$$\mathbf{T} = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

**RD**

2 3  $\rightarrow$  [2 3] (1)  
 0 3  $\rightarrow$  [0 4 3] (2)  
 0 2  $\rightarrow$  [0 1 2] (2)  
 3 4  $\rightarrow$  [3 4] (1)  
 2 4  $\rightarrow$  [2 1 4] (2)  
 0 1  $\rightarrow$  [0 1] (1)  
 1 4  $\rightarrow$  [1 4] (1)  
 1 2  $\rightarrow$  [1 2] (1)  
 1 3  $\rightarrow$  [1 2 3] (2)  
 0 4  $\rightarrow$  [0 4] (1)

**HoAS**

0 1  $\rightarrow$  [0 1] (1)  
 1 2  $\rightarrow$  [1 2] (1)  
 2 3  $\rightarrow$  [2 3] (1)  
 3 4  $\rightarrow$  [3 4] (1)  
 0 4  $\rightarrow$  [0 4] (1)  
 1 4  $\rightarrow$  [1 4] (1)  
 0 2  $\rightarrow$  [0 1 2] (2)  
 1 3  $\rightarrow$  [1 4 3] (2)  
 2 4  $\rightarrow$  [2 3 4] (2)  
 0 3  $\rightarrow$  [0 4 3] (2)

**HoAD**

0 1  $\rightarrow$  [0 1] (1)  
 0 4  $\rightarrow$  [0 4] (1)  
 1 2  $\rightarrow$  [1 2] (1)  
 1 4  $\rightarrow$  [1 4] (1)  
 2 3  $\rightarrow$  [2 3] (1)  
 3 4  $\rightarrow$  [3 4] (1)  
 0 2  $\rightarrow$  [0 1 2] (2)  
 0 3  $\rightarrow$  [0 4 3] (2)  
 2 4  $\rightarrow$  [2 1 4] (2)  
 1 3  $\rightarrow$  [1 2 3] (2)

**HoRD**

2 3  $\rightarrow$  [2 3] (1)  
 0 4  $\rightarrow$  [0 4] (1)  
 0 1  $\rightarrow$  [0 1] (1)  
 1 2  $\rightarrow$  [1 2] (1)  
 3 4  $\rightarrow$  [3 4] (1)  
 1 4  $\rightarrow$  [1 4] (1)  
 2 4  $\rightarrow$  [2 1 4] (2)  
 0 3  $\rightarrow$  [0 4 3] (2)  
 0 2  $\rightarrow$  [0 1 2] (2)  
 1 3  $\rightarrow$  [1 2 3] (2)

Fig. 1. Example of Ordering Heuristics.

$\mathbf{SP}(s, d) = h$  are added to  $T_{\text{list}}$  (line 8). This behavior allows to achieve a  $T_{\text{list}}$  comprising of all the non-zero elements in  $\mathbf{T}$ , or considering only the pairs of nodes with a specific route length ( $h$ ). To go through the  $p_n$  non-zero elements of  $\mathbf{T}$ , the time complexity of Algorithm 1 is estimated as  $O(p_n) = O(n^2)$ . Although when  $h \neq 0$ ,  $\text{Size}(T_{\text{list}}) < p_n$ , but we consider here only the worst case scenario where  $O(n^2)$ .

Fig. 1 illustrates the concept of getting  $\mathbf{NpO}$  with a practical example, considering a 5-node topology ( $n = 5$ ). Therefore, we explicitly show the  $\mathbf{SP}$  routes computed by using Dijkstra's algorithm for the  $p_n$  unique pairs of nodes, taking the connectivity matrix  $\mathbf{M}$  as an input. These routes

**Algorithm 2** Getting  $\mathbf{NpO}$  for RD heuristic**Require:**  $\mathbf{SP}, n$ 


---

```

1:  $T_{\text{list}} \leftarrow \text{GetListOfNodePairs}(\mathbf{SP}, n, 0)$ 
2: Function GetNpO4RD( $T_{\text{list}}$ )
3: while  $\text{Size}(T_{\text{list}}) > 0$  do
4:    $N_p \leftarrow \text{ChooseNodePair}(T_{\text{list}})$ 
5:    $\mathbf{NpO} \leftarrow N_p$ 
6:    $T_{\text{list}}.\text{removeElement}(N_p)$ 
7: end while
8: return  $\mathbf{NpO}$ 
9: end Function

```

---

are achieved for forward directions, and the backward ones are determined using the *Flip()* function. For all  $p_n = n(n-1)/2 = 10$  forward routes, we can see their lengths inside parenthesis. Therefore, we have 6 shortest path routes with length of 1 hop and 4 shortest path routes with length of 2 hops. The various versions of the  $\mathbf{NpO}$  matrix obtained with the proposed OHs are shown at the bottom of the figure, with the correspondent shortest path routes. They are obtained as follows:

1) *RD*: according to Algorithm 2, the dynamical list  $T_{\text{list}}$  with size  $p$  ( $h = 0$ ) is created, following Algorithm 1 (line 1). This list is the input of function *GetNpO4RD()* in line 2, where each one of its elements is a node pair ( $N_p$ ) represented by a two column array  $[s \ d]$ . Then, while there is at least one element in  $T_{\text{list}}$  (line 3), we randomly choose one element (node pair) from  $T_{\text{list}}$  (line 4). Therefore the two-column array  $[s \ d]$ , correspondent to  $N_p$ , is added at the end of  $\mathbf{NpO}$  bi-dimensional array (line 5) and removed from  $T_{\text{list}}$  (line 6). Taking  $\mathbf{NpO}$  achieved with RD ordering heuristic in Fig. 1, we can see that the length of routes is randomly distributed along  $\mathbf{NpO}$ . The time complexity of Algorithm 2 is estimated as  $O(p_n + p_n) = O(2p_n) = O(n^2)$ .

2) *HoAS*: this heuristic has a deterministic nature, and it does not depend on any random parameter. Following Algorithm 3, we alternate the source of node pairs in  $\mathbf{T}$  using ascending order of  $h$ . At first, the unique pairs of nodes whose  $\mathbf{SP}$ s have a length of 1 hop are taken, followed by those with  $\mathbf{SP}$ s of 2 hops, and so on. The maximum number of hops is known from the previous calculation of  $\mathbf{SP}$ s and it is called  $h_{\text{max}}$  (line 2). So, matrix  $\mathbf{T}$  is split in  $h_{\text{max}}$  sets. In our example of Fig. 1, it is split in two sets, generating a specific  $T_{\text{list}}$  for each of them (line 3). A control variable  $ss$  is set to zero (line 4), and it is incremented while  $T_{\text{list}}$  is not empty (line 5). The actual variable to represent source node  $s$  is achieved from the modulo operator between  $ss$  and  $n-2$ , producing an output that circularly runs from 0 to  $n-2$ , since  $\mathbf{T}$  has the  $n-1$  row with only zero elements (line 6). For each value of  $s$ , we look for the first occurrence of node pairs in  $T_{\text{list}}$ , whose first element  $s_i \in [s_i \ d]$  is equal to  $s$  (lines 7 to 9). If the element  $[s_i \ d]$  is found, it is added at the end of  $\mathbf{NpO}$  bi-dimensional array (line 10) and then removed from  $T_{\text{list}}$  (line 11). Once we have to change the source node  $s$ , the searching process must be interrupted (line 12) and the control variable  $ss$  is incremented by 1. Because of  $\mathbf{T}$  characteristic, we can not guarantee that destination node is kept the same

**Algorithm 3** Getting NpO for HoAS Heuristic

---

```

1: Function GetNpO4HoAS(T, SP,  $h_{\max}$ ,  $n$ )
2: for  $h \leftarrow 1, h_{\max}$  do
3:    $T_{\text{list}} \leftarrow \text{GetListOfNodePairs}(\text{SP}, n, h)$ 
4:    $ss \leftarrow 0$ 
5:   while  $\text{Size}(T_{\text{list}}) > 0$  do
6:      $s \leftarrow \text{modulo}(ss, n - 2)$ 
7:     for  $i \leftarrow 0, \text{Size}(T_{\text{list}}) - 1$  do
8:        $[s_l \ d] \leftarrow T_{\text{list}}.\text{getElementAt}(i)$ 
9:       if  $s_l = s$  then
10:         $\text{NpO}+ \leftarrow [s_l \ d]$ 
11:         $T_{\text{list}}.\text{removeElementAt}(i)$ 
12:        break
13:       end if
14:     end for
15:      $ss \leftarrow ss + 1$ 
16:   end while
17: end for
18: return NpO
19: end Function

```

---

when alternating among source nodes. Following our example and starting from  $T_{\text{list}}$  constructed with  $h = 1$ , the elements are  $[0 \ 1]$ ,  $[0 \ 4]$ ,  $[1 \ 2]$ ,  $[1 \ 4]$ ,  $[2 \ 3]$  and  $[3 \ 4]$ . After running this procedure, the sequence of elements stored in NpO is  $[0 \ 1]$ ,  $[1 \ 2]$ ,  $[2 \ 3]$ ,  $[3 \ 4]$ ,  $[0 \ 4]$  and  $[1 \ 4]$ . Then, the  $T_{\text{list}}$  for  $h = 2$  are achieved as  $[0 \ 2]$ ,  $[0 \ 3]$ ,  $[1 \ 3]$  and  $[2 \ 4]$ . Applying the described procedure, the elements  $[0 \ 2]$ ,  $[1 \ 3]$ ,  $[2 \ 4]$  and  $[0 \ 3]$  are appended, in this order, to NpO, as shown in Fig. 1. In order to calculate the time complexity of Algorithm 3, we overestimate the size of  $T_{\text{list}}$  as  $p_n$  for any specific value of  $h$ . Therefore, the time complexity of this algorithm is in overestimation  $O(h_{\max}(p_n + p_n^2)) = O(h_{\max}n^4)$ .

3) *HoAD*: another deterministic OH is HoAD. It follows the same idea of HoAS, with the difference that we alternate destination node  $d$  in the order it appears. Firstly we have to achieve  $T_{\text{list}}$  for each value of  $h$  (line 3). Then, we set two variables to store the last  $d$  node appended to NpO ( $d_a$ ), and to assure that all node pairs in  $T_{\text{list}}$  are appended to NpO( $ctrl$ ) (line 4). We search for node pairs while the size of  $T_{\text{list}}$  is greater than 1 (line 5). After that, we go through each element on it (line 6), taking the  $[s \ d]$  array (line 7) and testing if  $d$  is different from  $d_a$  or if  $ctrl$  is 1 (line 8). If any testing result is true, the element  $[s \ d]$  is added at the end of NpO bi-dimensional array (line 9) and then removed from  $T_{\text{list}}$  (line 10). In order to alternate the destination nodes, the next value of  $d_a$  is set to the current value of  $d$  (line 11) and the searching process must be interrupted (line 12). If the conditional statement in line 8 is not satisfied (line 13), we test if the last element in  $T_{\text{list}}$  was reached (line 14). If this results true,  $ctrl$  is set to 1 (line 15), relaxing the constraint imposed by line 8. This guarantees that every element on  $T_{\text{list}}$  is added to NpO even if alternate destination is not possible, due to **T** characteristic. This test is not necessary for HoAS, since in that procedure  $s$  runs circularly from 0 to  $n - 1$ , being possible to add elements with the same  $s$  in different cycles. Following our example of Fig. 1, for  $h = 1$  the elements

**Algorithm 4** Getting NpO for HoAD heuristic

---

```

1: Function GetNpO4HoAD(SP,  $h_{\max}$ ,  $n$ )
2: for  $h \leftarrow 1, h_{\max}$  do
3:    $T_{\text{list}} \leftarrow \text{GetListOfNodePairs}(\text{SP}, n, h)$ 
4:    $d_a \leftarrow -1, ctrl \leftarrow 0$ 
5:   while  $\text{Size}(T_{\text{list}}) > 0$  do
6:     for  $i \leftarrow 0, \text{Size}(T_{\text{list}}) - 1$  do
7:        $[s \ d] \leftarrow T_{\text{list}}.\text{getElementAt}(i)$ 
8:       if  $d \neq d_a$  or  $ctrl = 1$  then
9:         $\text{NpO}+ \leftarrow [s \ d]$ 
10:         $T_{\text{list}}.\text{removeElementAt}(i)$ 
11:         $d_a \leftarrow d$ 
12:        break
13:       else
14:        if  $i = \text{Size}(T_{\text{list}}) - 1$  then
15:           $ctrl \leftarrow 1$ 
16:        end if
17:       end for
18:     end for
19:   end while
20: end for
21: return NpO
22: end Function

```

---

**Algorithm 5** Getting NpO for HoRD heuristic

---

```

1: Function GetNpO4HoRD(SP,  $h_{\max}$ ,  $n$ )
2: for  $h \leftarrow 1, h_{\max}$  do
3:    $T_{\text{list}} \leftarrow \text{GetListOfNodePairs}((\text{SP}, n, h))$ 
4:    $\text{NpO}+ \leftarrow \text{GetNodePairsOrder4RD}(T_{\text{list}})$ 
5: end for
6: return NpO
7: end Function

```

---

in  $T_{\text{list}}$  are  $[0 \ 1]$ ,  $[0 \ 4]$ ,  $[1 \ 2]$ ,  $[1 \ 4]$ ,  $[2 \ 3]$  and  $[3 \ 4]$ , which are appended in this sequence to NpO, since the alternate destination constraint is obeyed. Switching to  $T_{\text{list}}$  for  $h = 2$ , the elements stored in  $T_{\text{list}}$  are  $[0 \ 2]$ ,  $[0 \ 3]$ ,  $[1 \ 3]$  and  $[2 \ 4]$ . Therefore, we append these elements to NpO, alternating destination, in the following order:  $[0 \ 2]$ ,  $[0 \ 3]$ ,  $[2 \ 4]$  and  $[1 \ 3]$ . The time complexity of Algorithm 4 is approximately the same as that of HoAS,  $O(h_{\max}(p_n + p_n^2)) = O(h_{\max}n^4)$ .

4) *HoRD*: The last OH proposed, HoRD, follows the same logic as HoAS and HoAD, concerning the different versions of  $T_{\text{list}}$ . The procedure is described in Algorithm 5, where firstly we have to go through the  $h$  values (line 2) and, for each one, achieve the specific  $T_{\text{list}}$  (line 3). Then, for the this list we apply function *GetNpO4RD*() shown in Algorithm 2 and append the output to NpO. The order of node pairs is randomly selected only inside the sets with the same route length. So, it is another OH with a random nature. An output example is also shown in Fig. 1. The time complexity of Algorithm 5 is estimated as  $O(h_{\max}(p_n + p_n)) = O(2p_n h_{\max}) = O(h_{\max}n^2)$ .

Other heuristics were tested, including hop-oriented in descending order. However, the results were worst or very close to the ones considered here. Thus, they are not presented in this work. It is important to highlight that only the adoption of an OH is not enough for a performance improvement.

Routing algorithms like Dijkstra's do not take in account the already calculated routes, or the future ones, to determine a given route between source and destination nodes. So, simply using these routing algorithms with the node pairs ordering is totally irrelevant. Nevertheless, the OH can introduce a meaningful improvement in load-balancing, and consequently a performance enhancement, if it is used along with a strategy to weight the link usage as explained in the following.

### B. Weighting Link Usage to get the Shortest Paths

Starting from the ordering strategy described above, the WOH proposed in this work is an iterative algorithm for the calculation of the SPs. In a very short description, the idea is to update the link costs at every SP calculation, so that the new SPs can take in account the usage of the links by new routes. The goal is clearly to have as much load balancing as possible, while still guaranteeing a shortest path between any pair of nodes. Therefore, routes achieved with this strategy are also shortest paths, and guarantee the symmetry of forward and backward directions. The backward route is calculated just after the forward one, as its inverse. For instance, Fig. 1 shows that route  $\text{SP}(0, 2)$  is  $[0 \ 1 \ 2]$  while  $\text{SP}(2, 0)$  is  $[2 \ 1 \ 0]$ .

Algorithm 6 illustrates the proposed strategy. It is supposed that we have a first version of SP, calculated by any routing algorithm for the unique pairs of nodes of  $\mathbf{T}$ . The function  $\text{Dijkstra}(\dots)$  returns, in time complexity  $O((n + e) \log n)$ , the shortest path route array depending on source ( $s$ ) and destination ( $d$ ) nodes, and on a weighted connectivity matrix  $\mathbf{M}_c$ . Furthermore, function  $\text{Length}()$  returns the length of a routing array. At the algorithm start-up, matrix  $\mathbf{M}$  is copied into matrix  $\mathbf{M}_c$  to preserve its content. The main structure of this algorithm is the loop to go through each forward node pair of NpO (line 2). All the new calculated routes must be a shortest path. Since it is not possible to change route lengths with 1-hop shortest paths, we can skip that situation with the conditional statement shown in line 3. Hence, given that a pair of nodes  $[s \ d]$  leads to  $\text{SP}(s, d) > 1$ , we calculate a testing route  $\mathbf{r}_t$  as function of  $s$ ,  $d$  and  $\mathbf{M}_c$  (line 4). If the lengths of testing and original SP routes are the same (line 5), the latter is replaced by the former (line 6). Therefore, we guarantee that the path between these 2 nodes is a shortest path route. Thus, if the test in line 5 returns false, the original route is not changed. The assignment in line 8 is used to assure that both forward and backward routes are symmetric. The loop structure starting on line 10 is used to update  $\mathbf{M}_c$  for both forward (line 11) and backward (line 12) routes between unique pair of nodes ( $s, d$ ), and it is executed even when the conditional statement in line 3 is not satisfied. So we have to go through each link in the given shortest path route, identifying the input and output nodes to perform the updates of  $\mathbf{M}_c$ . The links updating of Algorithm 6 is the key to understand the load balancing achievement, since route definition depends on the OHs. Furthermore, although very similar, HoAS and HoAD, can achieve different results in the definition of SPs because they act in a non symmetric way (while HoAS searches for pair of nodes in a circular sequence of  $s$ , HoAS searches for pair of nodes sequentially, trying to keep the same  $d$ ) Fig. 1 shows that for HoAS

### Algorithm 6 Getting SP with Weighted Ordering Heuristics

**Require:**  $\mathbf{M}, \text{NpO}, \text{SP}$

```

1:  $\mathbf{M}_c \leftarrow \mathbf{M}$ 
2: for all  $[s \ d] \in \text{NpO}$  do
3:   if  $\text{Length}(\text{SP}(s, d)) > 1$  then
4:      $\mathbf{r}_t(s, d) \leftarrow \text{Dijkstra}(s, d, \mathbf{M}_c)$ 
5:     if  $\text{Length}(\mathbf{r}_t(s, d)) = \text{Length}(\text{SP}(s, d))$  then
6:        $\text{SP}(s, d) \leftarrow \mathbf{r}_t(s, d)$ 
7:     end if
8:      $\text{SP}(d, s) \leftarrow \text{Flip}(\text{SP}(s, d))$ 
9:   end if
10:  for all  $\text{link} \in \text{SP}(s, d)$  do
11:     $\mathbf{M}_c(n_{\text{in}}^{\text{link}}, n_{\text{out}}^{\text{link}}) \leftarrow \mathbf{M}_c(n_{\text{in}}^{\text{link}}, n_{\text{out}}^{\text{link}}) + 1$ 
12:     $\mathbf{M}_c(n_{\text{out}}^{\text{link}}, n_{\text{in}}^{\text{link}}) \leftarrow \mathbf{M}_c(n_{\text{out}}^{\text{link}}, n_{\text{in}}^{\text{link}}) + 1$ 
13:  end for
14: end for
```

we have 2 routing changes (of node pairs  $[1 \ 3]$  and  $[2 \ 4]$ ) regarding original SPs, while for HoAD the routes remain the same. A hypothesis for this difference is the concentration of routes definition around a specific source node  $s$  for HoAD, alternating only the destinations. So, all the routes from  $s$  (and with a given  $h$ ) will be computed in sequence, loading the output links of  $s$ . In HoAS case, the  $s$  nodes alternate without keeping the same  $d$ . Thus, for each routing definition, the used links can be more distributed along the topology. Given that time complexity of Dijkstra's algorithm is  $O((n + e) \log n)$  in the best case, the conventional time complexity of getting all  $p_n$  forward routes is  $O(p_n(n + e) \log n) = O(n^3 \log n)$ . However, the time complexity using Algorithm 6 with HoAS or HoAD NpOs (that have the worst time complexities), is  $O(p_n(n + e) \log n + h_{\max} p_n^2 + (p_n - e)(n + e) \log n + p_n h_{\max}) = O(n^3 \log n + h_{\max} n^4 + n^3 \log n + h_{\max} n^2) = O(h_{\max} n^4)$ . Since the time complexities of HoAS and HoAD were overestimated, the overall time complexity using our proposed strategy, compared with the conventional one, is increased by approximately one order of  $n$ . Considering that all the routes are calculated just once and before the network booting, this difference between the time complexities has a very small impact in the performance of routing calculation.

The orientation to hop count is not a limitation and may be coupled with the adaptation of modulation in EONs to guarantee specific end-to-end link performance. For instance, given a certain fixed route (in hops), we can calculate the route distance (in km) and choose an acceptable modulation format to carry a specific bit rate with a required bandwidth, keeping the load balancing achieved by the presented strategy. Moreover, the basic ideas behind the WOH do not change if distance (in km) is used, even though a suitable quantitative tuning of the heuristic is probably needed. Algorithm 6 was tested with a traditional FGN (WDM) and with a EON.

## IV. PERFORMANCE MEASURES

### A. Load Normalization

In order to provide a fair comparison among topologies and network paradigms, we adopted a traffic load normalization

based on the idea of a theoretical maximum network capacity. This is called  $A_0$ , and can be calculated using topology parameters and the available amount of resources. The unit for  $A_0$  is erlang (E).

For the FGN it is simpler to compute  $A_0$ . We need the number of links in the given topology,  $e$ , and the amount of resources on each link, i.e. the total number of wavelengths,  $w$ . Given that an arbitrary connection uses just one wavelength out  $w$  and also a route made of one or more link(s) (hop), we can calculate the theoretical average number of hops  $\bar{h}$  for the topology, considering shortest path routes between every pair of nodes. Consequently the theoretical maximum network capacity can be defined as:

$$A_0 = \frac{w \cdot e}{\bar{h}}. \quad (2)$$

For the EON case, the definition of  $A_0$  is not so straightforward, because we do not know explicitly the amount of required resources for the connections. In fact, to the best of our knowledge, this metric is not discussed in literature and the formulation proposed here may be useful for other works. First of all, on each EON link, we have an available spectrum (total bandwidth),  $c$ . This spectrum is shared among a given set of services types  $S_t = \{s_{ti} \mid 1 \leq i \leq s_t\}$ . Therefore, each service  $s_i$  has two main parameters for the purpose of calculating  $A_0$ : (i) its bandwidth requirement,  $c_i$ , and (ii) its traffic probability,  $p_i$ . These parameters can be used to compute, taking all  $s_t$  services, an effective (mean) bandwidth requirement:

$$c_{\text{eff}} = \sum_{i=1}^{s_t} c_i p_i, \quad (3)$$

From  $c_{\text{eff}}$  at hand, we can assign an effective amount of resources as  $c/c_{\text{eff}}$ , that corresponds to  $w$  in (2) for the FGN case. This parameter can be used to compute  $A_0$  for the EONs:

$$A_0 = \frac{c \cdot e}{c_{\text{eff}} \bar{h}}. \quad (4)$$

Therefore, we can define the normalized network load  $a$ , which is dimensionless, ranging from 0 to 1. From  $a$  and  $A_0$ , the network traffic load,  $A$  (in erlang), can be calculated as:

$$A(a) = a A_0 \quad (5)$$

For instance, given a network whose maximum network capacity  $A_0$  is 1000 E, (5) establishes that a normalized network load of  $a = 0.5$  can be achieved with a network traffic load of 500 E.

## B. Performance Metrics

In this section we define some metrics used in this work. To do so, a generic EON will be considered, operating with  $s_t$  services and observed during a time  $t_{\text{ob}}$ . During  $t_{\text{ob}}$ , for each service  $s_{ti}$ , there is an amount of generated traffic  $g_i$ , and an amount of blocked traffic  $b_i$ . Thus, we can define the Blocking Probability,  $P_b$ , as:

$$P_b = \frac{\sum_{i=1}^{s_t} b_i}{\sum_{i=1}^{s_t} g_i}. \quad (6)$$

The average use of network resources is also a relevant metric. It is theoretically equals to 1 if all the established connections occupy all the resources over all the links during a period of time. Thus, taking an EON with  $e$  links in the network topology, each with  $r$  resources (wavelengths or spectrum), the average network utilization is given by:

$$U = \frac{\sum_{i=1}^{s_t} \sum_{g=1}^{g_i} \sum_{l=1}^e t_{\text{hold}}^{r'}(g, l)}{r \cdot e \cdot t_{\text{ob}}} \quad (7)$$

where  $t_{\text{hold}}^{r'}(g, l)$  is the time that generated connection  $g$  (of type  $i$ ) uses the resources  $r'$  (from the total  $r$ ) in the link  $l$ . It is important to note that in a EON,  $r'$  refers to the used spectrum slice, while in a FGN  $r'$  refers to the used wavelength. In this last case, the first sum disappears, since for FGNs we have only one kind of service. Furthermore, a connection will occupy a given set of links, that depends on the specific route, so the last sum is going through the used links.

Another aspect is that, due to continuity and contiguity constraints [4], longer routes tend to have higher blocking probabilities than shorter ones [18]. Therefore, we may want to study the blocking probability by route length. So, given that in a topology the longer routes have  $h_{\text{max}}$  hops (links), and that the route length  $h$  can be  $1 \leq h \leq h_{\text{max}}$ , the generation of connections with route length  $h$  is  $g_h$ , and blocked connections of length  $h$  is  $b_h$ . Therefore, we can define the blocking probability for a specific route length  $h$  as:

$$P_b^h = \frac{b_h}{g_h} \quad (8)$$

## V. NUMERICAL RESULTS

### A. Network Scenario

Our simulations were carried on using three frequently used network topologies, shown in Fig. 2:

- NSFNET, 14 nodes and 21 bidirectional links (42 directed ones), with  $h_{\text{max}} = 3$  and average number of hops  $\bar{h} = 2.143$ ;
- USANET, 24 nodes and 43 bidirectional links (86 directed ones), with  $h_{\text{max}} = 6$  and  $\bar{h} = 2.993$ ;
- PAN-EUR, 28 nodes and 41 bidirectional links (82 directed ones), with  $h_{\text{max}} = 8$  and  $\bar{h} = 3.561$ .

For the FGN paradigm we assumed 88 wavelengths, considering the ITU 50 GHz channel spacing, equivalent at having links with a total bandwidth of 4.4 THz (C-band). Using (2),  $A_0$  is 1724.8 E for NSFNET, 1149.3 E for USANET and 2026.5 E for PAN-EUR. For the EON case, we also have considered links with total capacity of 4.4 THz, frequency slots of 12.5 GHz, and  $s_t = 3$  services with the following bandwidth demands:  $c_1 = 25$  GHz,  $c_2 = 50$  GHz and  $c_3 = 100$  GHz, which include the FGN ITU channel with 50 GHz, its half bandwidth and its double bandwidth. For the sake of simplicity, we assume the bandwidth is the spectral resource which the system must reserve for the call, including any guard bands and/or overhead, if needed. Moreover, these bandwidth requirements match with some well-known modulation formats such as on-off keying (RZ or NRZ), differential quadrature phase shift keying (DQPSK), and OFDM. They

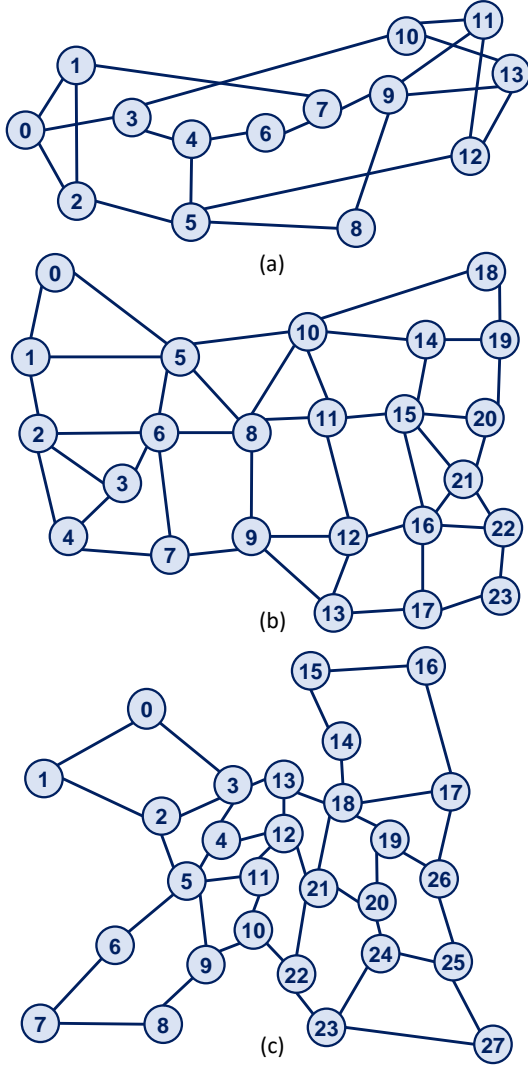


Fig. 2. (a) NSFNET, (b) USANET and (c) PAN-EUR Topologies.

can be considered to carry 10 Gb/s (with NRZ modulation format), 100 Gb/s and 400 Gb/s (using DP-QPSK modulation format) [1][26].

The traffic probabilities for EON services are inversely proportional to the bandwidth demands, following the formulation in [18], which means that services with higher bandwidth demands will have lower probabilities of being generated. This statement is reasonable, since we have much more traffic with lower bandwidth demands than with higher ones. For our specific case, the traffic probabilities were 57.1%, 28.6%, and 14.3%, respectively for services with bandwidth demands of 25, 50 and 100 GHz. Therefore, using (4), the maximum network capacity ( $A_0$ ) was calculated for NSFNET, USANET and PAN-EUR topologies as, respectively, 2012.3, 2950.2 and 2364.3 E. It is important to mention that, given our load normalization method, any set of services or traffic probabilities can be used to carry out an EON performance analysis, keeping the overall qualitative results.

For both FGN and EON, the connection arrivals follow a Poisson distribution with average  $\lambda$  and the connection dura-

TABLE I  
ROUTING CHANGES, BY LENGTH, FOR NSFNET

Parameters	NSFNET topology		
$h$	1	2	3
#Routes	42	72	68
#RC RD*	0	2	9
#RC HoAS	0	3	8
#RC HoAD	0	2	11
#RC HoRD*	0	2	9

\*Average of 30 replications rounded to the nearest integer.

TABLE II  
ROUTING CHANGES, BY LENGTH, FOR USANET

Parameters	USANET topology					
$h$	1	2	3	4	5	6
#Routes	86	134	138	106	68	20
#RC RD*	0	12	23	24	14	3
#RC HoAS	0	15	22	25	13	4
#RC HoAD	0	21	26	25	15	3
#RC HoRD*	0	9	22	23	15	5

\*Average of 30 replications rounded to the nearest integer.

tions follow a negative exponential distribution with average  $1/\mu$  equal to 10 time units. One million connections were generated for each normalized network load point, ranging from 0.05 to 1.0, with intervals of 0.05. With this number of connections we can guarantee a statistical confidence higher than 95%. Also, we concentrated the analysis in the window between 0.2 up to 0.7, since it gives the most interesting values of blocking probability, ranging from 0.001 to 0.1. Furthermore, for RD and HoRD OHs, we made 30 replications, since the orders defined in  $NpO$  are expected to change depending on the seeds of the random number generator. These 30 seeds were chosen as prime numbers, distributed from 3 to 997. The mean, considering the 30 replications, is used to trace the curves of RD and HoRD, but for these cases we show also the error bars for a 95% confidence interval on each point.

Wavelength (FGN) or spectrum (EON) resources were assigned with the first-fit heuristic, which chooses always the first available option. Results of blocking probability and network utilization concerning WOH fixed-routing strategy were compared with an alternate-routing strategy using the  $k$ -shortest paths ( $k$ -SP), with  $k$  from 1 to 3. Although this strategy tends to increase the route lengths, it was important to compare our propose with a known effective approach that improves the network performance [11].

### B. Performance Analysis

As stated before, to deal with a statistically significant number of data for RD and HoRD, simulations were repeated 30 times. To understand how the proposed WOH affects the chosen routes, at first we counted the number of routes that change as a result of applying it. Table I, Table II and Table III show the results for the three adopted topologies. In all these tables, the row marked as the #Routes indicates the number of routes for  $h = 1, 2, 3$  or more hops as set up

TABLE III  
ROUTING CHANGES, BY LENGTH, FOR PAN-EUR TOPOLOGY

Parameters	PAN-EUR topology							
$h$	1	2	3	4	5	6	7	8
#Routes	82	142	162	152	118	64	30	6
#RC RD*	0	8	19	24	25	16	9	2
#RC HoAS	0	9	18	25	29	20	11	2
#RC HoAD	0	15	19	28	29	18	8	2
#RC HoRD*	0	6	19	26	27	19	11	3

\*Average of 30 replications, rounded to nearest integer.

TABLE IV  
STATISTIC FOR THE NUMBER OF SUPPORTED ROUTES, BY LINK

Routing Order	Topologies								
	NSFNET			USANET			PAN-EUR		
Heuristic	AVG	STD	CV	AVG	STD	CV	AVG	STD	CV
1-SP	9.29	3.49	37.6	19.21	10.89	56.7	32.83	20.34	61.9
RD*	9.29	2.57	27.6	19.21	7.92	41.2	32.83	14.15	43.1
HoAS	9.29	2.48	26.7	19.21	8.02	41.7	32.83	14.14	43.1
HoAD	9.29	2.87	30.9	19.21	7.94	41.3	32.83	14.22	43.3
HoRD*	9.29	2.50	26.9	19.21	7.90	41.1	32.83	14.03	42.8

\*Average of 30 replications.

by conventional shortest path routing (1-SP). The following rows (marked as #RC) indicate the number of routes that changed when the specific WOH routing policy was applied. No changes are observed for the 1-hop routes. However, all the considered strategies impose some route changes for the routes that involve 2, 3 or more hops. For instance, for HoAS in the NSFNET, 3 out of 72 routes with 2 hops are changed. This represents a total of 4.2% of these connections. For the 3-hops situation, this number increases to 11.8% ( $= 8/68$ ). It is important to note that for larger topologies, as USANET and PAN-EUR, the number of routing changes is more significant. For instance, the number of routing changes for HoAS and  $h = 2$  is 15 for USANET and 9 for PAN-EUR, which represents 11.2% and 6.3% of these connections, respectively.

Table IV presents the statistics for the number of supported routes by each link. Since the WOH does not affect the route length, the average number of supported routes per single link is the same with and without WOH, meaning that, using the WOH strategy, we guarantee that our fixed routes are also shortest paths. For instance in the NSFNET this average (AVG) is calculated as  $(1 \cdot 42 + 2 \cdot 72 + 3 \cdot 68)/42 \approx 9.29$ . The differences that affect the network performance depends on the standard deviation (STD) of this statistic. If the numbers of supported routes by link were uniformly distributed, which means that each link would support the same number of routes, the STD would be zero. However, the uneven link distribution among nodes in a given mesh topology and the search for shortest paths make difficult to achieve a uniform distribution of supported routes by link. Some links will support more routes than other ones, and these differences can be estimated by the STD. It means that, if we have a high value of STD, probably we have higher number of links supporting much more (and much less) routes than the

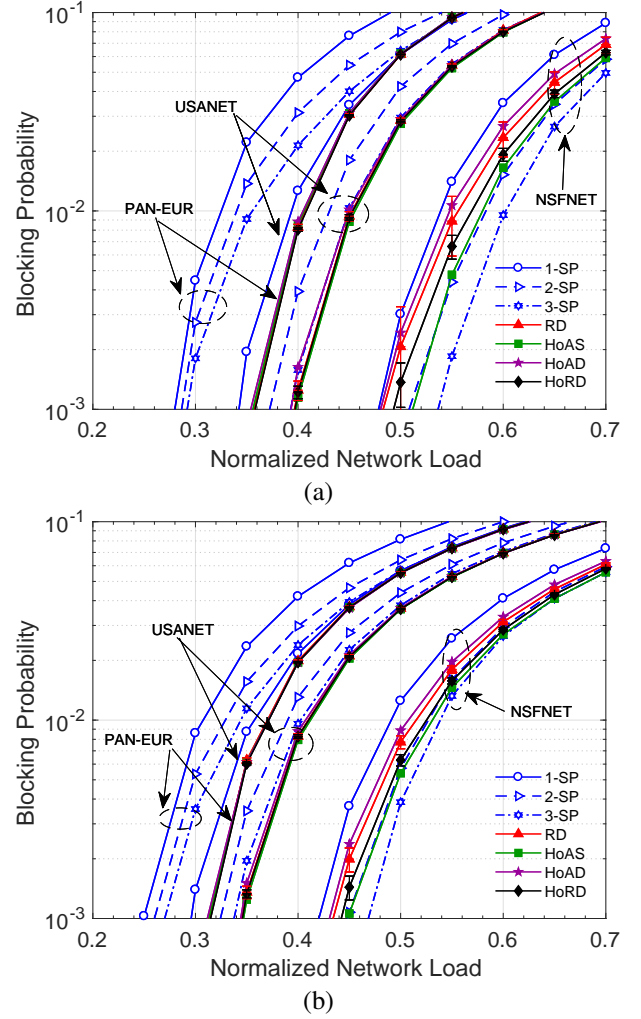


Fig. 3. Blocking Probability for (a) FGN and (b) EON.

average. On the other hand, if we have lower values of STD, we can conclude that we have a more balanced distribution of supported routes by link. From Table IV if we use a fixed route without using WOH (1-SP), the STD is 3.49 for NSFNET, 10.89 for USANET and 20.34 for PAN-EUR, respectively. These numbers lead to values of Coefficient of Variation (CV), calculated as  $100 \cdot (\text{STD}/\text{AVG})$ , of 37.6%, 56.7% and 61.9%, respectively for NSFNET, USANET and PAN-EUR. Using, for instance, WOH HoAS, the values of CV are 26.7%, 41.7% and 43.1%, respectively for NSFNET, USANET and PAN-EUR. These last values are interesting because they are lower than the values verified for 1-SP case. Therefore, CV can be used as a metric to previously evaluate a topology regarding the use of some WOH strategy. Furthermore, the differences in CV comparing WOH strategies for USANET and PAN-EUR were very little, while for NSFNET they are perceptible.

The most significant performance figure, the blocking probability, is presented in Fig. 3 for all WOH fixed-routing strategies and for  $k$ -SP, with  $k$  up to 3. In almost all considered situations the proposed WOH fixed-routing outperformed even the 3-SP conventional approach. The exception was for the smaller NSFNET topology, where the performance of the best WOH, HoAS, is quite similar to 2-SP, but it is outperformed by

3-SP. However, all WOHs outperform 1-SP for this topology. A hypothesis for this behavior is the small size of NSFNET, compared with the other ones. The smaller the topology, the lower the probability of finding a different shortest path route. In fact, Fig. 1 shows that only HoAS heuristic was able to change routes for that 5-nodes topology. Furthermore, it is interesting to note that the performance of WOHs follow the tendency of CV values of NSFNET (Table IV), for both FGN and EON cases. The study provided by Table IV does not make sense for 2-SP and 3-SP, since for these routing strategies we can have longer routes, increasing the average number of supported routes by link (AVG). Considering NSFNET and comparing 1-SP and HoAS for  $P_b = 1\%$ , the normalized network load may be increased from 0.54 to 0.56 (FGN) or from 0.49 to 0.53 (EON), leading to 3.7% and 8.1% of performance improvement, respectively. These results are even more significant if we take USANET and PAN-EUR topologies. For USANET, comparing 1-SP and HoAS, results suggest that network operators may improve their performance by an amount of 15.1% (FGN) and 18.3% (EON). Using PAN-EUR, the performance may be improved by 26.1% (FGN) and 20.7% (EON). Moreover, it is important to highlight that all WOHs applied to USANET and PAN-EUR topologies outperformed even 3-SP for FGN and EON.

Another issue regarding blocking probabilities among WOHs is that for NSFNET, they are noticeably different, while for USANET and for PAN-EUR they are practically equal. This behavior is also supposed to be explained from Table IV, where CV for RAND, HoAS, HoAD and HoRAND are, respectively, 27.6%, 26.7%, 30.8% and 26.9%, which are perceptively different. Analyzing the CV column data for USANET, these percentages are, respectively, 41.2%, 41.7%, 41.3% and 41.1%, which are clearly much closer among them. It also happens with PAN-EUR topology. Analyzing the network paradigm, it can be noted that, compared to FGN, EON presents a higher level of blocking probability for lower loads, but a lower blocking level for higher loads. To explain this behavior we have to remember about the requiring bandwidths for the FGN (50 GHz) and for EON (25 GHz, 50 GHz and 100 GHz). Compared to NSFNET with 1-SP and FGN, the performance of EON case is worse in lower loads because of a combination of the spectrum fragmentation and of the unfairness in loss probability among different types of service. For 1 million connections, the number of blocked ones for  $a = 0.5$  with FGN was 3017. For the same scenario, the total number of blocked connections for EON was 12512, being 28, 1421 and 11063, respectively for services demanding bandwidths of 25 GHz, 50 GHz and 100 GHz. This behavior evidences that dealing with services with a higher bandwidth demand is more difficult, even for lower loads. However, for higher loads the spectrum gets busier and also more fragmented, becoming even harder to setup connections with bandwidth demand of 100 GHz. On the other hand, connections with bandwidth demand of 25 GHz (that also have the highest arrival rate in our scenario) may be more easily accommodated. Consequently, the number of blocked calls for  $a = 0.7$  with FGN was 88573, while for EON it was 73347, being 957, 18374 and 54016, respectively for

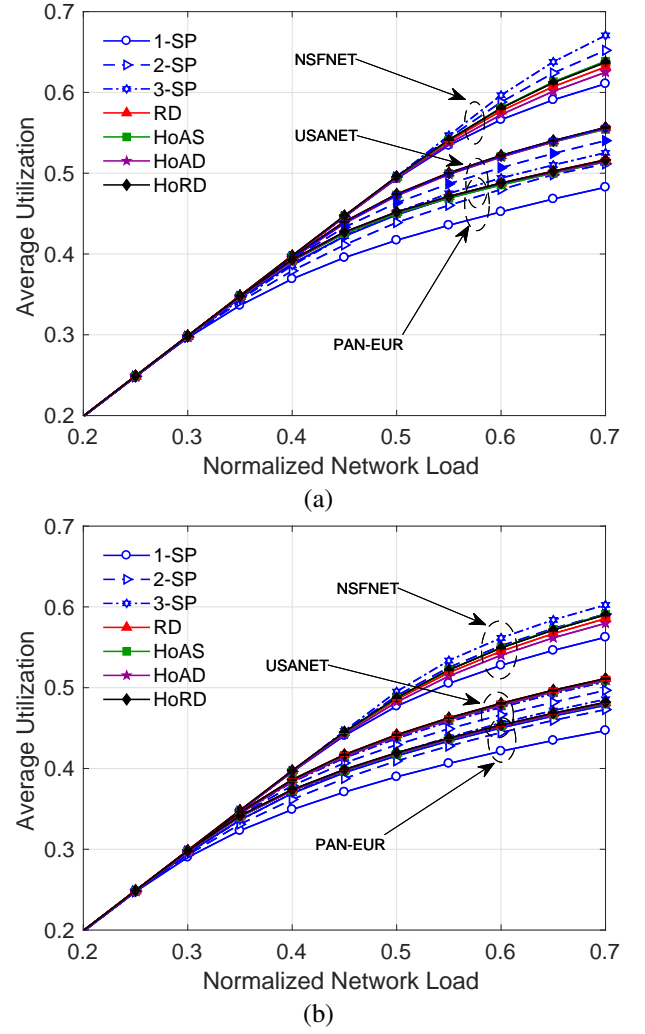


Fig. 4. Average Network Utilization for (a) FGN ( $s = 1$  in (6)) and (b) EON.

services demanding bandwidths of 25 GHz, 50 GHz and 100 GHz. It is intuitively understandable, since in higher loads the bandwidth available on the links may not be enough to accept services with more demanding bandwidths, whereas it can be sufficient to accommodate services with lower demands [26].

Once we normalized the load, using (2) and (4), to reach a fair comparison among all analyzed scenarios, the average utilization, evaluated from (7), is very important. Fig. 4 shows the results for the average utilization, which help us to validate the proposed normalization, since theoretically, the average utilization should be exactly equal to the normalized network load (a). Therefore, we can see that for lower loads, the average utilization, as function of normalized network load, is a line supposedly crossing the origin with angular coefficient equal to  $45^\circ$ . Due to network dynamics and topology nature, from a given value of  $a$  the blocking probability increases due to the lack of resources and spatial fragmentation (for FGN case), making the average utilization to decrease. Besides that, in EON scenario, the spectral fragmentation and the unfairness problem are other issues that contribute for the decreasing of average utilization. So, if we analyze a given topology with a certain WOH (and even the case we do not use the WOH

strategy), the average utilization in the FGN case is always higher.

Furthermore, network utilization is enhanced when the proposed WOH strategies are utilized. This behavior is a consequence of the better load distribution supplied by these strategies and of the resulting lower blocking probability. Interestingly, all considered WOH strategies roughly provide the same network utilization for USANET topology whereas HoAS WHO strategy presents the best performance for NSFNET topology. These results hold for both FGN and EON scenarios and, again, track the blocking probability trends presented in Fig. 3. Another issue is that network utilization tends to be higher as the  $k$  value of  $k$ -SP increases, which is expected, since we are using longer routes and more network resources.

Finally, Fig. 5 shows the results of blocking probability by route length for a EON with  $a = 0.5$ . The important message here is that the WOH always reduces the blocking probability experienced by long routes. The more the reduction, the longer the routes. For instance the longest routes in the PAN-EUR network have 8 hops and WOH almost halves the blocking probability on these routes as an effect of a better load balancing. We avoided this analysis for 2-SP and 3-SP alternate-routing, since these strategies increase the route lengths.

Summarizing the previous results, we presented a WOH strategy that led to

- the adoption of some routes that are different from the ones encountered by the traditional approach (without WOH), but that are also shortest paths;
- a better load balancing of network link usage;
- a network utilization improvement;
- a blocking probability reduction for routes encompassing any number of hops;
- an enhancement of overall network performance with a decrease in blocking probability, even compared to  $k$ -SP alternate-routing approach.

Furthermore, another characteristic to highlight is that the computational cost for this strategy is low, and there is no need of additional network devices.

## VI. CONCLUSION

In this manuscript we have proposed a new Weighted Ordering Heuristics (WOH) to static route determination in fix-grid and elastic optical network scenarios. The WOH was tested on three topologies widely used in scientific studies and compared with conventional alternate-routing technique  $k$ -SP. Results shows that the WOH may enhance the performance of the network, increasing the network utilization and decreasing the blocking probability. This performance improvement relies on the implementation of new ordering heuristics, which requires very limited computational effort, and demands no installation of extra hardware or software. The numerical results also show that the proposed strategy automatically leads to the establishment of new shortest path routes and that routes encompassing any number of hops experienced a blocking probability decrease. Interestingly, this reduction is enhanced for longer routes (i.e., involving a higher number of hops).

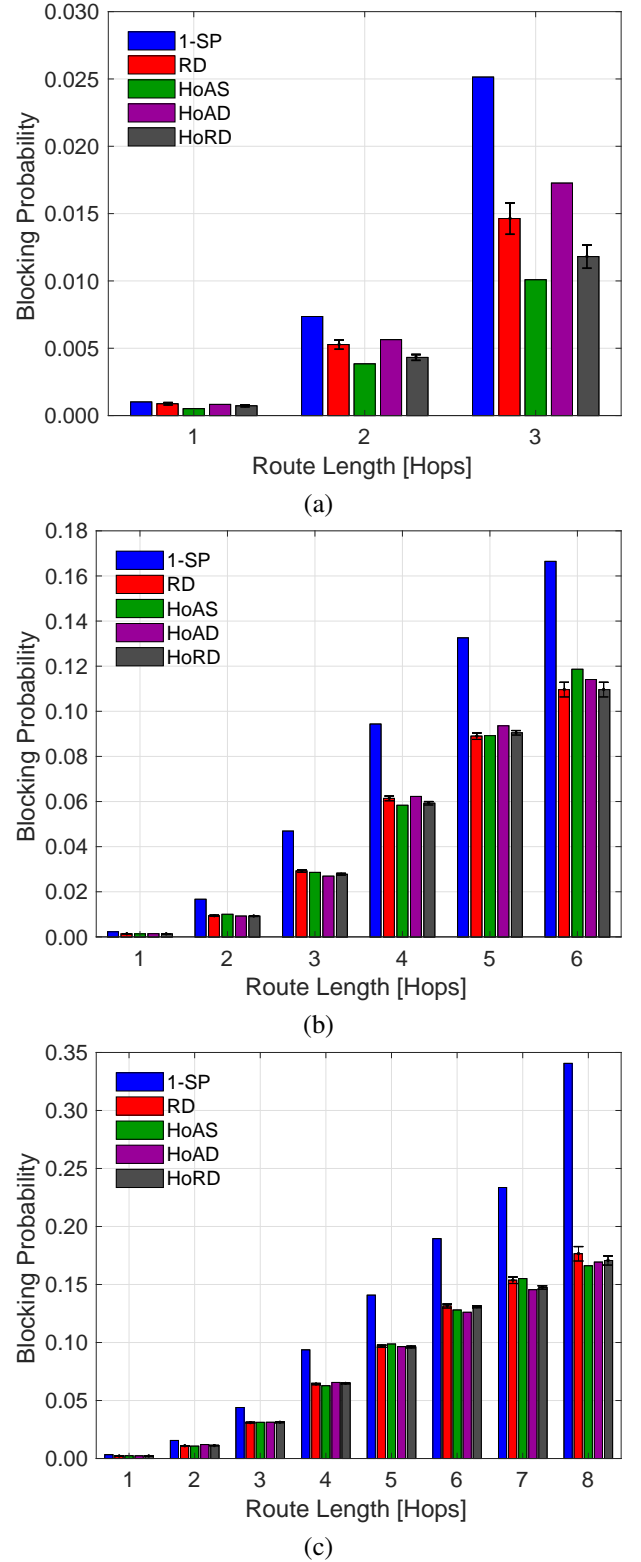


Fig. 5. Blocking Probability by Hop Length with  $a = 50\%$  for EON and (a) NSFNET, (b) USANET and (c) PAN-EUR topologies.

## ACKNOWLEDGMENT

The authors thank the Brazilian agencies FAPESP (grants 2008/57857-2, 2016/23511-9 and 2015/24341-7), CNPq (grants 574017/2008-9, 311137/2014-8, 305427/2016-4) and CAPES (Finance Code 001) for the financial support. The

authors also thank Prof. Denise Hideko Goya and Dr. Majid Forghani-elahabad for the discussions on time complexity analysis.

## REFERENCES

- [1] O. Gerstel, M. Jinno, A. Lord and S. J. B. Yoo, "Elastic optical networking: a new dawn for the optical layer?", *IEEE Communications Magazine*, Vol. 50, no. 2, pp. 12–20, Feb. 2012.
- [2] Y. Yoo, S. Ahn and C. S. Kim, "Adaptive routing considering the number of available wavelengths in WDM networks," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 8, pp. 1263–1273, Oct. 2003.
- [3] X. Wan, L. Wang, N. Hua, H. Zhang, and X. Zheng, "Dynamic Routing and Spectrum Assignment in Flexible Optical Path Networks," *2011 Optical Fiber Communication Conference/National Fiber Optic Engineers Conference 2011 (OFC/NFOEC)*, paper JWA055, 2011.
- [4] M. Klinkowski and K. Walkowiak, "Offline RSA algorithms for elastic optical networks with dedicated path protection consideration", *IV International Congress on Ultra Modern Telecommunications and Control Systems*, pp. 670–676, 2012.
- [5] H. Zang, J. P. Jue and B. Mukherjee, "A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks", *Optical Networks Magazine*, no. 1, pp. 47–60, Jan 2000.
- [6] B. C. Chatterjee, N. Sarma and E. Oki, "Routing and Spectrum Allocation in Elastic Optical Networks: A Tutorial," *IEEE Communications Surveys & Tutorials*, Vol. 17, No. 3, pp. 1776–1800, thirdquarter 2015.
- [7] H. Lin, S. Wang, and M. Hung, "Finding Routing Paths for Alternate Routing in All-Optical WDM Networks," *J. Lightwave Technol.*, Vol. 26, 1432–1444, 2008.
- [8] H. Harai, M. Murata and H. Miyahara, "Performance of alternate routing methods in all-optical switching networks", *17th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'97)*, Vol. 2, pp. 516–524, 1997.
- [9] B. Chatterjee, N. Sarma, and P. Sahu, "Priority Based Routing and Wavelength Assignment With Traffic Grooming for Optical Networks," *J. Opt. Commun. Netw.*, Vol. 4, pp. 480–489, 2012.
- [10] L. H. Bonani, M. Forghani-elahabad, "An improved least cost routing approach for WDM optical network without wavelength converters," *Optical Fiber Technology*, Vol. 32, pp. 30–35, 2016.
- [11] K. Walkowiak, A. Kasprzak, M. Klinkowski, "Dynamic Routing of Anycast and Unicast Traffic in Elastic Optical Networks", in *Proceedings of the IEEE International Conference on Communications (ICC)*, pp. 3313–3318, Aug. 2014.
- [12] P. H. Ho and H. T. Mouftah, "An Approach for Enhancing Fixed Alternate Routing in Dynamic Wavelength-Routed WDM Networks", in *Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE*, vol. 3, pp. 2792–2797, 2002.
- [13] G. Shen, S. K. Bose, T. H. Cheng, C. Lu and T. Y. Chai, "Performance analysis under dynamic loading of wavelength continuous and non-continuous WDM networks with shortest-path routing", *Int. J. Commun. Syst.*, vol. 14, pp. 407–418, 2001.
- [14] A. Agrawal, V. Bhatia, S. Prakash, "Spectrum efficient distance-adaptive paths for fixed and fixed-alternate routing in elastic optical networks," *Optical Fiber Technology*, Vol. 40, pp. 36–45, Jan. 2018.
- [15] R. Ramamurthy and B. Mukherjee, "Fixed-alternate routing and wavelength conversion in wavelength-routed optical networks," *IEEE/ACM Transactions on Networking*, Vol. 10, No. 3, pp. 351–367, Jun. 2002.
- [16] H. C. Lin, S. W. Wang and C. P. Tsai, "Traffic Intensity Based Fixed-Alternate Routing in All-Optical WDM Networks," *IEEE International Conference on Communications (ICC)*, pp. 2439–2446, 2006.
- [17] E. W. Dijkstra, "A note on two problems in connexion with graphs", *Journal of Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, Dec. 1959.
- [18] L. H. Bonani, A. S. Tozetti, F. Callegati and W. Cerroni, "Routing Issues on Spectrum Sharing and Partitioning for Flexible Optical Networks", *Proceedings of the 2014 16th International Conference on Transparent Optical Networks (ICTON)*, paper Th.B3.3, pp. 1–4, Jul. 2014.
- [19] G. Duraes, A. Soares, J. R. Amazonas and W. Giazza, "Performance evaluation of fixed routing algorithms in all-optical networks under non-uniform traffic", in *Proceedings of the 2010 Fifth International Conference on Systems and Networks Communications*, pp. 368–374, Aug. 2010.
- [20] P. Rajalakshmi and A. Jhunjunwala, "Load Balanced Routing to Enhance the Performance of Optical Backbone Networks, in *Proceedings of 5th IFIP Int. Conf. on Wireless and Optical Communications Networks (WOCN 2008)*, Surabaya, Indonesia, 2008.
- [21] G. Duraes, A. Soares, J. R. Amazonas and W. Giazza, "The choice of the best among the shortest routes in transparent optical networks", *Computer Networks*, vol. 54, pp. 2400–2409, 2010.
- [22] Y. Li, L. Peng and G. Shen, "Load-Balanced Fixed Routing for Wavelength Routed Optical Networks", *IEEE Communications Letters*, vol. 17, no. 6, pp. 1256–1259, Jun 2013.
- [23] G. Shen, Y. Li, L. Peng, "A novel load-balanced fixed routing (LBFR) algorithm for wavelength routed optical networks", *Proceedings SPIE 8331, Photonics and Optoelectronics Meetings (POEM) 2011: Optical Communication Systems and Networking*, 833109, Jan. 2012.
- [24] Y. Jungang, N. Junyong and T. Xiangyuan, "A load balance based routing and wavelength assignment algorithm", in *Proceedings of the 2014 7th International Conference on Intelligent Computation Technology and Automation*, pp. 23–27, Oct 2014.
- [25] C. K. Constantinou and G. Ellinas, "A load balancing technique for efficient survivable multicasting in mesh optical networks", *Optical Switching and Networking*, vol. 22, pp. 1–8, Nov. 2016.
- [26] F. Callegati, L. H. Bonani and W. Cerroni, "Service Fairness in Flexible Optical Networks", *2013 Optical Fiber Communication Conference and Exposition and the National Fiber Optic Engineers Conference (OFC/NFOEC)*, OTH1H.4, Mar. 2013.