Optical Fiber Technology 50 (2019) 99–107

Contents lists available at ScienceDirect

Optical Fiber Technology

journal homepage: www.elsevier.com/locate/yofte

A machine learning approach to mitigating fragmentation and crosstalk in space division multiplexing elastic optical networks

Yu Xiong[a,b,c,*], Yaya Yang[a,b,c], Yulong Ye[a,b,c], George N. Rouskas[d,e]

[a] School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, China
[b] Key Laboratory of Optical Communication and Network in Chongqing, China
[c] Key Laboratory of Ubiquitous Sensing and Networking in Chongqing, China
[d] Department of Computer Science, North Carolina State University, Raleigh, NC, USA
[e] Department of Computer Science, King Abdulaziz University, Jeddah, Saudi Arabia

ARTICLE INFO

Keywords:
Space division multiplexing elastic optical networks
Spectrum fragmentation
Machine learning
Crosstalk

ABSTRACT

As network traffic is expected to continue to grow at high rates for the foreseeable future, it becomes imperative to introduce space division multiplexing elastic optical networks (SDM-EONs) into the optical transport network. However, spectrum fragmentation and crosstalk present significant challenges that may negatively impact the performance of SDM-EONs. In this paper, we leverage machine learning techniques to enhance the transmission performance of SDM-EONs, and make two contributions. Specifically, we use an Elman neural network to forecast traffic demands, and use a two-dimensional rectangular packing model to allocate spectrum so as to decrease unnecessary spectrum fragmentation (and, in turn, increase resource utilization). We also present a novel spectrum partition scheme to reduce crosstalk. Our evaluation study confirms that the proposed strategy is effective in improving spectrum utilization while reducing blocking probability and crosstalk.

1. Introduction

With the proliferation of bandwidth-intensive services, including high-definition video and cloud computing, and the swift emergence of 5G high-speed mobile media services and Internet of Things applications, traffic on backbone networks has increased drastically and is expected to continue on a steep upwards trajectory. As a result, traffic demands may soon bump into the capacity limits of elastic optical networks (EONs) [1]. Space division multiplexing (SDM) technologies, including multi-fiber, -core, and -mode, are currently under extensive investigation [2] in an effort to accommodate more network traffic and prevent such an undesirable outcome.

EONs allocate spectrum at a fine granularity that matches the spectrum requirements of each request [1]. Nevertheless, as network traffic varies over time, spectrum tends to become fragmented due to the setup and release of lightpaths [3], leading to a decrease in resource utilization and a decline in performance. SDM technology based on multi-core fiber (MCF) expands the amount of available spectrum resources so as to accommodate additional requests, which in turn leads to a larger amount of wasted resources in SDM-EONs for a given degree of fragmentation.

In order to mitigate the impact of spectrum fragmentation, a variety of strategies have been studied recently that may generally be classified as either *reactive* (i.e., those employing defragmentation) or *proactive* (i.e., those that attempt to prevent fragmentation). Reactive strategies are triggered either periodically or whenever fragmentation exceeds a certain threshold, so as to reroute some or all of the in-service connections onto new lightpaths. In [4], the authors proposed methods for intelligent timing selection and adaptive defragmentation ratio selection to tackle the tradeoff between bandwidth blocking probability performance and operational complexity. In this study, connection reconfigurations were carried out sequentially, an approach that introduces significant latency. Parallel defragmentation, proposed in [5], was shown to be effective in reducing latency by conducting all connection reconfigurations simultaneously. However, extensive traffic disruption may occur at defragmentation instants. To address this issue, the authors of [6] proposed a preemptive reconfiguration strategy that only reroutes a portion of existing connections at a time so as to minimize traffic disruption.

Clearly, reconfiguration always incurs traffic disruption to some degree, hence it increases delay and deteriorates the performance of the network. Therefore, the research community has also investigated proactive strategies that attempt to avoid or minimize spectrum fragmentation by employing appropriate routing and spectrum allocation

* Corresponding author at: School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, China.
E-mail address: cqxiongyu@foxmail.com (Y. Xiong).

https://doi.org/10.1016/j.yofte.2019.03.001
Received 9 December 2018; Received in revised form 16 February 2019; Accepted 2 March 2019

algorithms at the time connection requests arrive. The strategy proposed in [7] splits a request into multiple sub-requests and routes each along a different path so as to utilize fragmented spectrum resources across the network. Nevertheless, it is well known that such reverse multiplexing of connections increases operating costs as it requires complex control and management. The work in [8] defined a new fragmentation ratio to characterize various routing and spectrum assignment solutions, and presents fragmentation-aware algorithms that allocate paths and spectrum to connection requests so as to keep the fragmentation ratio low. However, the above representative studies as well as many others, assumed a resource model that does not take the time dimension into account. The work in [9] proposed a two-dimensional resource model for each link that accounts not only for the spectrum demands but also the holding time of each connection. This model represents the spectrum resources on each link over time and may be used to make more effective routing and spectrum allocation decisions for each request. The authors of [10] introduced the concept of spare spectrum availability to accommodate connection requests. Based on it, two different crosstalk-aware routing, spectrum, and core assignment algorithms are presented at different network states, which improves the spectrum resource utilization and reduces blocking, with a cost of additional algorithm complexity and set-up delay. A node-arc-based integer linear programming formulation was designed in [11] to assign the lightpath, the cores, and the corresponding spectrum simultaneously for each transmission request, two XT-aware-based approaches that consider XT strictly also be proposed, fulfilling all the requisite constraints, such as inter-core crosstalk and spectrum overlapping. However, it is suitable for static RSCA problem.

Most proactive strategies in the literature, including the ones discussed above, were designed for optical networks with single-core fiber. The SDM-EONs with multi-core fiber (MCF) that we consider in this work are susceptible to signal impairments due to inter-core crosstalk [12]. Hence, crosstalk should be taken into consideration in the process of resource allocation. The work in [13] introduced an on-demand strategy that takes both crosstalk and spectrum fragmentation into consideration in assigning spectrum resources. Specifically, crosstalk is reduced by avoiding the allocation of the same frequency slots in adjacent cores, whereas fragmentation is avoided by having each core serve connections with the same demands, to the extent possible. The results show that the strategy is effective in reducing both network blocking probability and crosstalk.

The premise of our work is that by anticipating future traffic requests, it is possible to apply appropriate algorithms to reduce spectrum fragmentation and crosstalk, hence improve network performance in terms of spectrum utilization and blocking probability. To this end, we make three contributions. First, we use machine learning techniques, and specifically the Elman neural network (ENN), to obtain an accurate forecast of future traffic based on historical data. Second, we present a resource model for spectrum allocation that represents the time dimension and captures crosstalk constraints among adjacent cores of a fiber. Finally, we develop a parameterized machine learning-assisted fragmentation avoidance (MLFA) algorithm that leverages information on predicted traffic from the ENN and the resource model to address the dual challenges of spectrum fragmentation and crosstalk.

The rest of this paper is organized as follows. Following this introduction, in Section 1 we present the network model we consider in this work. In Section 2 we introduce a traffic prediction model based on the ENN, and in Section 3 we develop a resource model for spectrum allocation in multi-core fiber. We present our routing, spectrum and core assignment algorithm for fragmentation avoidance in Section 4 and evaluate it in Section 5. We conclude the paper in Section 6.

2. Network model

We consider an online version of the routing, core, and spectrum assignment (RCSA) problem in SDM-EONs of arbitrary topology in which each link consists of a single multi-core fiber (MCF), whereby the cores provide the space dimension for switching. We assume that each node is equipped with a spatially and spectrally resolved optical switching fabric as shown in Fig. 1 [14]. The fabric comprises a SDM

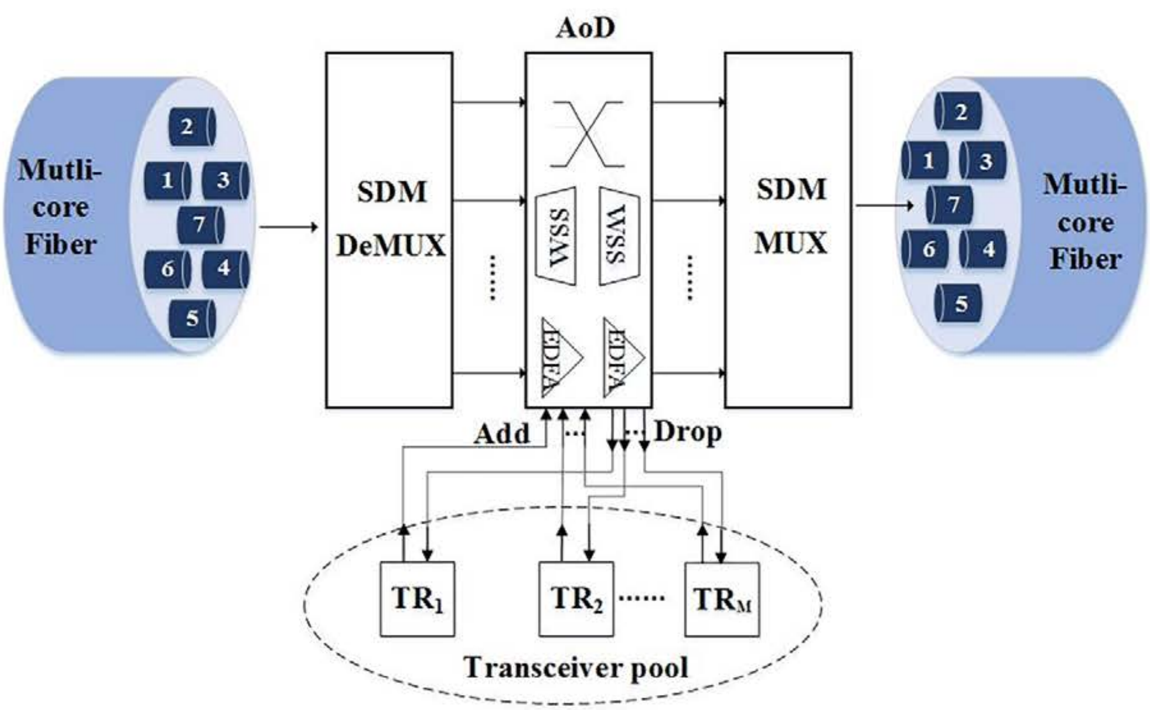


Fig. 1. Spatially and spectrally resolved optical switching fabric for SDM-EONs.

Demultiplexer (DeMUX), an architecture on demand (AoD) module, and a SDM Multiplexer (MUX). The SDM DeMUX and SDM MUX are used to interconnect the MCFs terminating at the node.[1] The AoD module includes bandwidth-variable wavelength selective switches (WSS) and erbium-doped fiber amplifiers (EDFA), allowing the adding, dropping, and switching of flexible channels at the granularity down to one spectrum slot. Each node also includes a transceiver (TR) pool from which appropriate sub-transceivers are allocated to each connection depending on its traffic demand.

Fig. 2(a) shows a seven-core MCF, the most common structure that we adopt in this work as illustrated in Fig. 1. An MCF has multiple cores and achieves a far larger transmission capacity than a traditional single-core fiber. However, crosstalk arises as a major challenge in MCF. Specifically, as shown in Fig. 2, crosstalk occurs when optical signals using the same spectrum propagate within adjacent cores in the MCF. In other words, when the same spectrum slots are occupied by live traffic on adjacent cores, inter-core crosstalk is generated [15]. Inter-core crosstalk leads to serious physical layer impairment and degrades the quality of service, thus the crosstalk should be avoided during allocation which results with higher spectrum fragmentation. It is possible to allocate lower number of requests, which results with higher blocking probability [16]. In addition, the introduction of multiple cores adds a new dimension of complexity with respect to spectrum assignment, potentially leading to more severe fragmentation of the spectrum resources.

We assume that connection requests arrive and depart dynamically, and each request is characterized by three parameters: (1) the size Φ of the request in units of Gbps; (2) the arrival time t_a; and (3) the holding time t_h. We assume that the bandwidth of each spectrum slot is 12.5 GHz. Once the network receives a new request, it runs an RCSA algorithm to assign a path, contiguous spectrum, and core to the request. In the remainder of this paper we describe our approach to designing such an algorithm for SDM-EONs.

3. Traffic prediction based on machine learning

Machine learning techniques are widely used for prediction and decision-making, hence it is no surprise that recently such techniques have been introduced in the operation and design of optical networks; for a comprehensive survey we refer the reader to [17]. In particular, three recent studies are closely related to our work. The authors of [18] proposed deep learning-based techniques to improve traffic prediction accuracy in optical data center networks for resource allocation. Machine learning technology has also been used to predict the crosstalk in SDM-EONs [19], while back propagation neural networks were used to forecast the size of future traffic requests in [20].

The Internet traffic exhibits a similar pattern, the similar or periodic traffic pattern over a long period has been referred to as time-varying traffic. Therefore, traffic prediction is feasible. In this section our goal is to predict not only the size Φ of future demands as was done in our work [20], but also their arrival time t_a and holding time t_h, which are time-varying. In Elman neural network (ENN), there is no need to use state variable as the input or training signal and its dynamic characteristic is provided by its internal connection, which makes this network more suitable for time-varying system modeling. This is also an important factor which makes ENN superior to the back propagation neural network in [20]. To this end, we apply the ENN model [21]. Our goal is to leverage accurate traffic prediction to inform resource allocation in SDM-EONs so as to avoid spectrum fragmentation and crosstalk.

Fig. 3 [21] shows the ENN model for traffic forecasting. It consists of four layers: input layer, hidden layer, context layer and output layer [2] each described in detail next. The hidden layer neurons are fed by the outputs of the context neurons and the input neurons. Context neurons are known as memory units as they store the previous output of hidden neurons. This network structure makes the network more sensitive to the historical data, which increases its ability to deal with the dynamic information.

Input layer. In the input layer, the node output $x_i(k)$ is the same as the node input net_i (i.e., there is no transformation of the input):

$$x_i(k) = net_i, \quad i = 1, 2, \tag{1}$$

where k represents the k-th iteration.

Hidden layer. In the hidden layer, the transfer function is a sigmoid function $S(x)$. The node input and node output are represented as:

$$net_j = \sum_i w_{ij} x_i(k) + \sum_r w_{rj} x_r(k), \quad r = 1, 2, \ldots, 9, \tag{2}$$

$$x_j(k) = S(net_j) = \frac{1}{1 - e^{-net_j}}, \quad j = 1, 2, \ldots, 9. \tag{3}$$

In the above expressions, $x_r(k)$ is the output of the context layer, and net_j, $x_j(k)$ represent the input and output of the hidden layer, respectively. Furthermore, w_{ij} (respectively, w_{rj}) refer to the weights of the links that connect the input layer to the hidden layer (respectively, the context layer to the hidden layer).

Context layer. In the context layer, the node input and the node output are expressed as:

$$x_r(k) = x_j(k - 1). \tag{4}$$

In other words, the output of the context layer at iteration k is the output of the hidden layer at the previous iteration $k - 1$.

Output layer. In the output layer, the node input and the node output are calculated by the following two formulas

$$net_o = \sum_j w_{jo} x_j(k) \tag{5}$$

$$y_o(k) = f_i(net_o(k)) = net_o(k) \tag{6}$$

Note that w_{jo} are the weights of links between the hidden and output layers, and $y_o(k)$ represents the output of the ENN.

The prediction error E is obtained from the expected output $y_o(k)$ of the network and the actual value $y_o'(k)$:

$$E = \sum_{k=1}^{n} [y_o'(k) - y_o(k)]^2 \tag{7}$$

The error may be reduced by continually adjusting the values of weights w_{ij}, w_{rj}, and w_{jo} during the training process until the error does not exceed a predefined threshold ξ. Specifically, the weights are updated as follows:

$$w_{jo}(k + 1) = w_{jo}(k) + \eta\left[-\frac{\partial E}{\partial y_o}\frac{\partial y_o}{\partial net_o}\right]\left(\frac{\partial net_o}{\partial w_{jo}}\right) \tag{8}$$

$$w_{rj}(k + 1) = w_{rj}(k) + \eta\left[-\frac{\partial E}{\partial y_o}\frac{\partial y_o}{\partial net_o}\right]\left(\frac{\partial net_o}{\partial w_{jo}}\frac{\partial x_j}{\partial w_{rj}}\right) \tag{9}$$

$$w_{ij}(k + 1) = w_{ij}(k) + \eta\left[-\frac{\partial E}{\partial y_o}\frac{\partial y_o}{\partial net_o}\right]\left(\frac{\partial net_o}{\partial x_j}\frac{\partial x_j}{\partial w_{ij}}\right) \tag{10}$$

where η is the learning rate.

A pseudocode description of the training process of ENN model is provided as Algorithm 1. The input e_i to the algorithm represents the attributes [Φ, t_a, t_h] (i.e., size, arrival time, and holding time) of a set of historical traffic requests. As a first step, the training data and initial

[1] Although Fig. 1 fig:switch shows a degree-2 node, nodes in the SDM-EON may be of higher degree, in which case they are equipped with appropriately extended switch fabrics.

[2] We use subscripts i, j, r, o to denote quantities related to the input, hidden, context and output layers, respectively, in expressions (1)–(10) below.

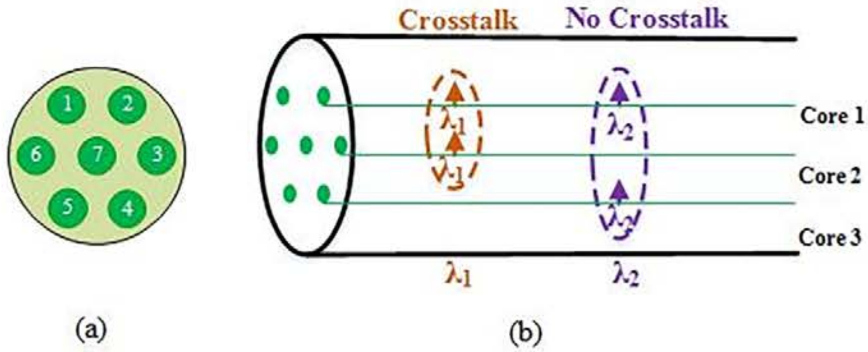


Fig. 2. (a) Structure of the MCF, (b) crosstalk in MCF.

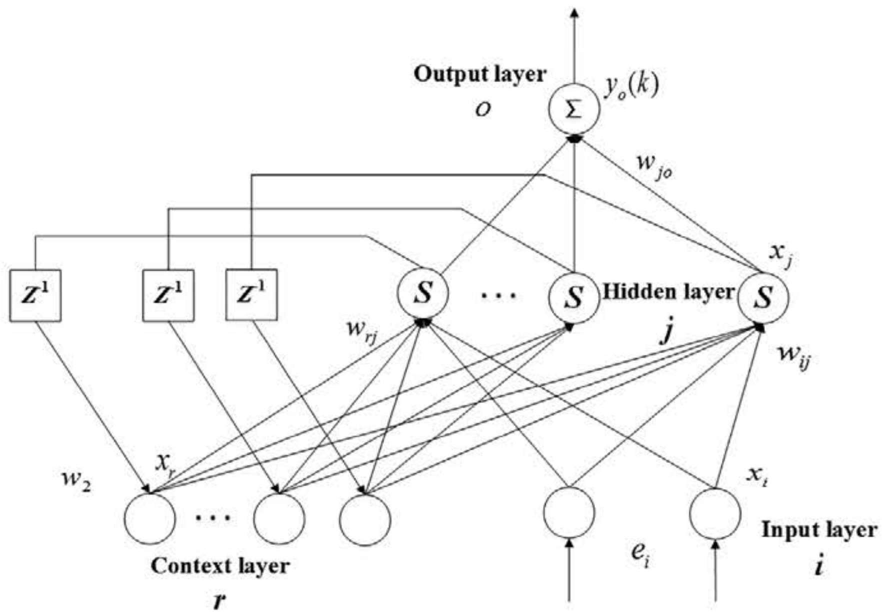


Fig. 3. Architecture of the Elman neural network.

weights w_{ij}, w_{rj}, w_{jo} are input (Lines 1–2). The ENN training process follows in Lines 3–13. The input and output of all layers are calculated in Line 6 according to formulas (1)–(6), and in Line 5 the error E is calculated as in expression (7). If the error is below the predefined threshold ξ, then the prediction results $y_o(k)$ are produced in Line 7. Otherwise, the weights are updated in Line 8 following expressions (8)–(10) and the process is repeated until the error decreases below the threshold.

Algorithm 1 The training process of ENN model

Input: ENN structure, e_i
1: Input the training dataset: $[\Phi, t_a, t_h]$ of traffic requests
2: Initialize the weights w_{ij}, w_{rj}, w_{jo}
3: Set the learning rate 0.001
4: batch learning on training dataset
5: **for** each batch **do**
6: Calculate the input and output of all the layers according to forward propagation
7: Calculate the error function E
8: **if** $E < \xi$ **then**
9: Return the prediction results $y_o(k)$: $[\Phi, t_a, t_h]$
10: **else**
11: Update the weights w_{ij}, w_{rj}, w_{jo} until $E < \xi$
12: **end if**
13: **end for**

To evaluate the ENN model, we used data from the Center for Applied Internet Data Analysis website (CAIDA)[22] over a two-week period. The number of our training set is 100,000 requests. A small batch gradient descent method is used to train the neural network, each batch is 100, and an Adam optimizer is used. We used the first-week traffic as historical data to train the ENN, and applied the ENN model to predict traffic over the second week. The dotted line in Fig.4 represents the actual two-week traffic data from [22], whereas the solid line represents the traffic over the second week as predicted by the ENN model. As indicated in Fig.4, the ENN model accurately captures future behavior based on the past traffic profile; the error in this prediction is kept below 5%. In the following, we leverage the traffic prediction capabilities of ENN to develop a spectrum allocation scheme that avoids fragmentation.

4. Resource models for multi-core fiber

The dynamic setup and release of connections is bound to fragment the available spectrum on each core into small non-continuous pieces that may not be used to accommodate future requests. If not appropriately addressed, spectrum fragmentation may severely impact network performance, including in terms of spectrum utilization and blocking probability. In this section, we propose two complementary resource models designed to alleviate fragmentation and crosstalk in

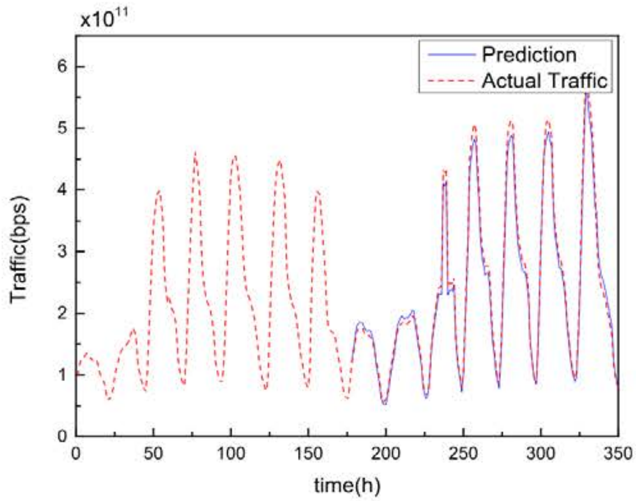


Fig. 4. The actual traffic and prediction using ENN model.

SDM-EONs with MCF. The first model applies to spectrum within a single core, while the second takes into consideration crosstalk between signals in adjacent cores. Using these models, in the following section we develop a spectrum allocation algorithm for fragmentation avoidance. Such allocation is subject to the continuity and contiguity constraints.

4.1. Crosstalk-aware inter-core resource allocation

The introduction of MCF into SDM-EONs allows for significant expansion of carrying capacity proportional to the number of cores in each fiber. Nevertheless, an important side effect of having multiple cores in the same fiber is the inter-core crosstalk generated when light signals occupy the same spectrum slots in adjacent cores [12]. Inter-core crosstalk leads to serious physical layer impairments and degrades the quality of service; consequently, the result is an increase in the number of connections rejected, and, hence, in blocking probability [14]. The statistical mean inter-crosstalk of a homogenous of a MCF per unit length is expressed in formula (11). Furthermore, by considering the coupled-power theory, the crosstalk of a MCF can be expressed as in formula (12) [23].

$$h = \frac{2k^2 r}{\beta D} \tag{11}$$

$$XT = \frac{N - N\cdot\exp[-(N-1)\cdot 2hl]}{1 + N\cdot\exp[-(N-1)\cdot 2hl]} \tag{12}$$

Formula (11) denotes the mean increase in inter-core crosstalk per unit length, where k, r, β, D are the relevant fiber parameters, respectively the coupling coefficient, bend radius, propagation constant, and core-pitch. XT is the mean crosstalk. N is the number of the adjacent cores, and l is the fiber length.

We now present a spectrum partitioning scheme that aims to avoid inter-core crosstalk. Consider two adjacent cores, C_i and C_j, $i \neq j$, and let S denote the total amount of spectrum available in each core. We partition the spectrum S of the two cores into two contiguous areas, S_{i1}, S_{i2} and S_{j1}, S_{j2}, as shown in Figure 5, so that the first (respectively, second) area of core C_i has no overlap with the second (respectively, first) area of core C_j in terms of spectrum slots. Specifically, the sizes of the two areas in each core are calculated as:

$$S_{i1} = S_{j1} = \alpha_t S \tag{13}$$

$$S_{i2} = S_{j2} = (1 - \alpha_t)S \tag{14}$$

where α_t is a parameter that is adjusted over time as discussed shortly.

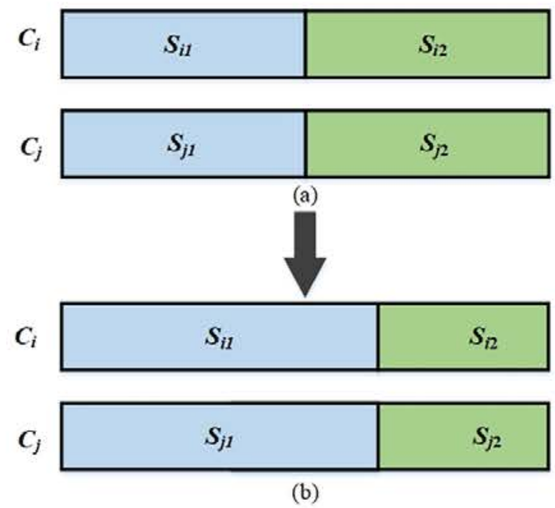


Fig. 5. Spectrum partitioning: (a) initial state, (b) final state.

With this partitioning scheme, we adopt the following spectrum allocation policy:

For two adjacent cores C_i and C_j, when sub-area S_{i1} (respectively, S_{j2}) is used for carrying traffic, then new connections on core C_j (respectively, C_i) may be allocated spectrum only from sub-area S_{j2} (respectively, S_{i1}).

As the amount of traffic carried by each core changes dynamically over time, it is important to adjust the sizes of the two spectrum sub-areas accordingly to ensure that the cores can serve more future requests and blocking probability is reduced. Therefore, we update the value of α_t in response to changes in the relative traffic load carried (i.e., amount of spectrum used) by cores C_i and C_j. Let H_{it} and H_{jt} be the traffic amounts carried by cores C_i and C_j at time t, respectively. Then, α_t is calculated as:

$$\alpha_t = \frac{H_{it}}{H_{it} + H_{jt}}. \tag{15}$$

As a result, the size of the each spectrum sub-area in each core is adjusted as connections get established and terminated. For instance, the release of a connection when the spectrum in the two cores is in the state depicted in Fig. 5(a) will result in an increase (respectively, decrease) of the first sub-areas S_{i1} and S_{j1} (respectively, second sub-areas S_{i2} and S_{j2}) as shown in Fig. 5(b) due to an adjustment in the value of α_t.

As a final note, the center core (i.e., core C_7 in the MCF depicted in Fig. 2(a)) is a special case as it is adjacent to all other six cores. To reduce overlap of used spectrum between C_7 and the other cores, we allocate spectrum on this core randomly.

4.2. Horizon-based intra-core spectrum allocation

We abstract the resources on each core of each link as a two-dimensional (2D) time-spectrum pool, as shown in Fig. 6. The x axis in the figure represents the available spectrum (in units of slots corresponding to a bandwidth $B = 12.5$ GHz), whereas the y axis represents available time(in units of slots corresponding to a time slot $\tau = 10$ min). The time horizon T in Fig. 6 represents how far into the future resources will be allocated (reserved) to accommodate connection requests that are expected to arrive later. The time dimension is also divided in non-overlapping time slots which represent the unit of time for which spectrum is allocated. The four solid color rectangles in Figure 6 represent four connections that have already been allocated spectrum within the specific 2D time-spectrum pool.

Each connection request $[\Phi, t_a, t_h]$ may be represented as a rectangle of area $M = f \times n_{th}$, where n_{th} is the number of time slots required and f

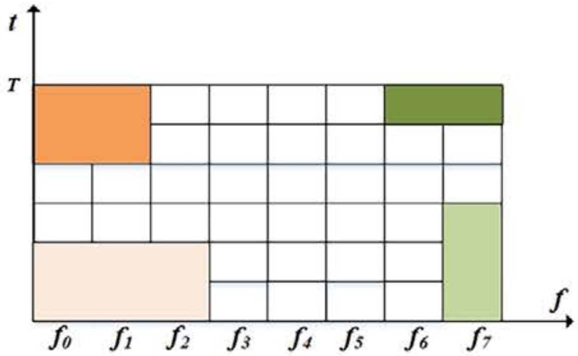


Fig. 6. 2D time-spectrum pool.

is the number of spectrum slots required. t_h is the holding time of the connection. The modulation format we consider in this work is BPSK, if modulation level L is determined, then the number of spectrum and time slots is obtained as follows:

$$f = \frac{\Phi}{B \times \log_2 L} \tag{16}$$

$$n_{th} = \left\lceil \frac{t_h}{\tau} \right\rceil \tag{17}$$

Given a number of connection requests that are forecast to arrive within the time horizon T (i.e., requests with $t_a \leqslant T$), the problem of allocating spectrum on a specific core of a specific link in the network becomes equivalent to a 2D rectangular packing problem [23,24]. In addition, a request can be served on only one core and respects the contiguity and continuity of spectrum. In 2D rectangular packing, the objective is to place rectangular items on a predefined rectangular area so that no items overlap and the area covered by the placed items is maximized. With this insight, the spectrum allocation problem is simplified and we can build upon results from 2D rectangular packing theory. In particular, while the 2D rectangular packing problem has been shown to be is NP-hard, several heuristics have been developed [24].

In this work, we use a two-step algorithm for 2D rectangular packing that incorporates elements from the algorithms in [23,24]:

- *Step 1.* Sort all connection requests in decreasing order of the area M of the corresponding rectangle, and allocate spectrum to each in this order.
  Note that rectangles with a large area M require many contiguous time and spectrum slots to be placed. On the other hand, there is more flexibility in placing small rectangles as they may fit into empty areas even if the time-spectrum pool has been fragmented. Therefore, placing large rectangles first is expected to achieve higher resource utilization.
- *Step 2.* For each request, use the *fit degree* defined in [22] to select one of possible multiple potential placements for the corresponding triangle. The objective is to minimize fragmentation of the time-spectrum pool.

Since small requests are allocated after large ones, it is possible that there are several fragments within the time-spectrum pool large enough to place the corresponding rectangles. The fit degree was introduced in [24] to rank the various placements of small rectangles, and it was shown that selecting the placement with largest *fit degree* leads to better resource utilization. Due to page constraints, we omit the definition of the *fit degree*, and refer the reader to [24] for all the details.

5. Resource allocation algorithm for SDM-EONs with MCF

The routing, core, and spectrum allocation algorithm for SDM-EONs is shown as Algorithm 2, and leverages both the ENN model prediction and the 2D rectangular packing algorithm; we refer to Algorithm 2 as machine learning-assisted fragmentation avoidance (MLFA) strategy.

Algorithm 2: Machine Learning-Assisted Fragmentation Avoidance (MLFA)

Input: network topology $G(V, E)$, requests $r_i \in R$
Out: Routing, core and spectrum assignment (RCSA) solution (or blocked request)
1: **for** $r_i(s, d, \Phi, t_a, t_h)$ **do**
2: $\Phi \leftarrow$ENN model
3: $t_a \leftarrow$ENN model
4: $t_h \leftarrow$ENN model
5: compute k-shortest path between s-d
6: **for** each path **do**
7: select an available core C on each link according to First-Fit
8: divide the slots on C according to the partition scheme in Section 4.1
9: **end for**
10: **for** ($t_a \leqslant T$) **then**
11: calculate the M
12: sort r_i and stored them in set P in descending order of M
13: **end if**
14: **for** each r_i(in decreasing value of M) **do**
15: allocate resource according to 2D rectangular packing algorithm in Section 4.2
16: if no resource is available, block the request
17: **end for**
18: return RCSA solution or block the request
19: **end for**

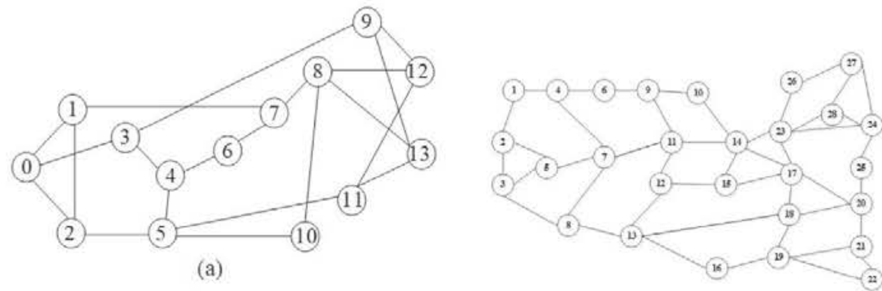
As a first step, MLFA applies the ENN model to forecast traffic demands (Lines 1–4), which are determined by their source node, destination node, size, arrival time and holding time. k shortest paths from the source s to the destination d of the connection are calculated. For each candidate shortest path, the core on each link of each path is selected for the request according to the well-known First-Fit policy (Lines 5–7).The spectrum slots of the core are then divided into two sub-areas, S_1 and S_2, as described in Section 4.2 (Line 8). If a request cannot be established on a given path, check the second path, and then the next one, until the request can be allocated. If it is not possible then rejected the request.

Following this initialization, the algorithm then attempts to allocate resources for all requests with an arrival time $t_a \leqslant T$. Specifically, all these requests are first sorted and stored in set P in decreasing order according to the area M of their corresponding rectangle (Lines 10–13). Then, each request is considered in this order (Lines 14–17), and spectrum slots are allocated using the rectangular packing algorithm we described in Section 4.2. Spectrum slots are assigned starting with the first of the k paths until one is found that may accommodate the connection. If a path with available resources exists, the corresponding RCSA solution is returned; otherwise, if none of the k paths may accommodate the request, the connection is blocked.

6. Numerical results

6.1. Simulation setup

We developed a C++ simulation tool to evaluate the proposed MLFA strategy. For the experiments, we used two topologies: the 14-node, 21-link NSFNET and the 28-node, 45-link USNET shown in Fig. 7(a) and 7(b), respectively. The number of candidate routes is $k = 3$, the bandwidth of each spectrum slot is assumed to be 12.5 GHz [25]. Each time slot is 10 min [26]. All links have a single MCF with 7 cores as shown in Fig. 2(a), and each core has 320 spectrum slots. Connection requests arrive following a Poisson process, and the holding time of each request is assumed to follow a negative exponential distribution. The traffic load is given by the utilization of the bottleneck link, in which the most lightpaths are assigned when all source-



(a) NSFNET (b) USNET

Fig. 7. Network topologies: (a) NSFNET, (b) USNET.

destination pairs have the same arrival rate [13]. We set the value of the time horizon to $T = 12$ hrs, as in [9]. The algorithm runs 500 times each time, and the results are obtained from 100,000 requests, which confidence interval is 95%, with less than 5% error for each displayed point of diagram.

In the experiments, we compare four strategies:

1. *First-Fit*: a baseline strategy that allocates available spectrum slots and cores such that the indices of the allocated slots are minimized.
2. *On-demand*: the strategy in [13], which applies two predefined policies. The first is a core selection priority for reducing crosstalk, and the second is a core classification for reducing spectrum fragmentation.
3. *MLFA-WT*: our proposed MLFA strategy but *without* traffic prediction. Specifically, when a request arrives, resources (path, core, spectrum slots) are allocated for it immediately based on the current network resource state information. In other words, the request is immediately filled into the 2D time-spectrum pool.
4. *MLFA*: the strategy we described in the previous section.

The data for the ENN model were obtained from [22]. Also, we set the number of nodes in the hidden layer of the ENN to 9, and the learning rate to $\eta = 0.001$ [27].

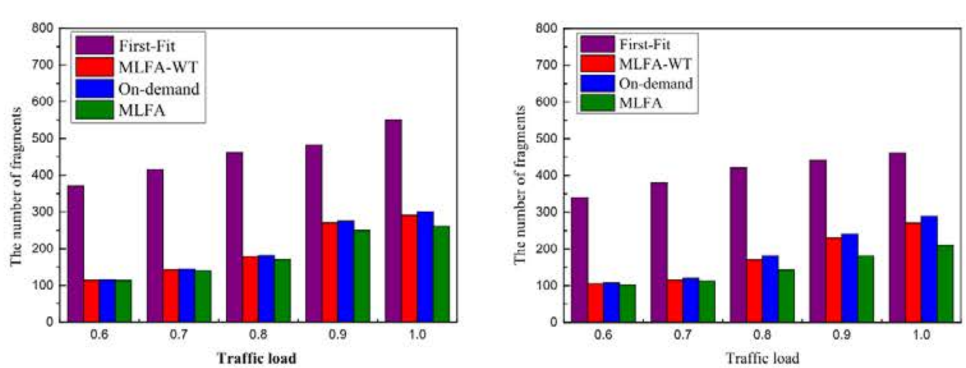
We compare the various algorithms with respect to four metrics:

- Average number of one- or two-slot fragments in the entire network.
- Spectrum utilization (SU) of the entire network, defined as the percentage of slots occupied by existing requests among all the slots in the fiber links.
- Crosstalk per slot (CpS), defined as the ratio of the arrangement of spectrum slots that generate crosstalk to the number of used slots.
- Blocking probability (BP) of connections across the entire SDM-EON.

6.2. Simulation results

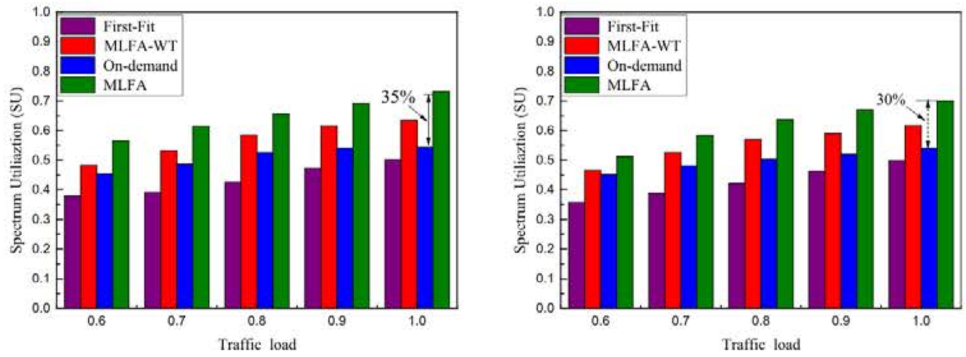
Let us first consider the average number of one- or two-slot fragments in SDM-EONs. Fig. 8 plots the average number of one- and two-slot fragments as a function of the traffic load, for the NSFNET topology. In our simulation, we assume that connections request at least three spectrum slots, hence one- and two-slot fragments cannot be used to serve any requests and represent wasted resources. As shown in the figure, the First-Fit strategy results in significantly more spectrum fragmentation than other strategies. This is attributed to the fact that First-Fit is a greedy policy that always allocates the first (lowest-index) available set of slots, without any attempt to utilize other parts of the spectrum, and doing so may create holes that may not accommodate future requests. The MLFA-WT scheme performs slightly better than the On-demand strategy, indicating that the 2D rectangular packing model indeed is effective in allocating resources. Note also that the MLFA-WT and MLFA strategies generate roughly the same fragmentation at low loads. But at higher loads when resources are scarce, MLFA clearly outperforms MLFA-WT. This result is mainly due to the fact that MLFA prioritizes connection requests according to the size and holding time predicted by the ENN model, whereas MLFA-WT serves requests on a first-come, first-served basis. In other words, in the absence of traffic prediction, the spectrum allocation scheme is only locally optimal. On the other hand, horizon-based allocation allows for an overall planning solution for requests during the horizon T, and hence traffic prediction leads to a result that is closer to the global optimum. Consequently, MLFA makes it possible to reduce spectrum fragmentation significantly.

Fig. 9 plots the spectrum utilization (SU) against the traffic load for the NSFNET and USNET topologies. For the reasons discussed above, the First-Fit strategy has the lowest SU among the four strategies we considered in our study. The On-demand strategy utilizes spectrum more efficiently than First-Fit, mainly due to the predefined core classification policy [13]. Specifically, connections requiring the same



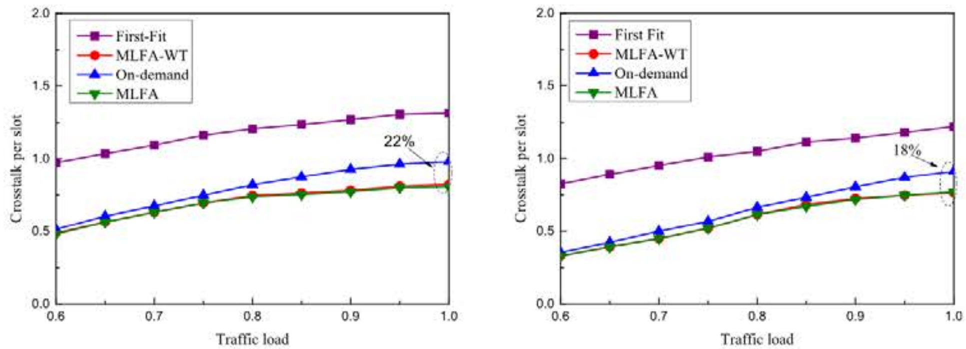
(a) NSFNET (b) USNET

Fig. 8. Number of fragments: (a) one-slot, (b) two-slot.



(a) NSFNET (b) USNET

Fig. 9. Spectrum utilization: (a) NSFNET, (b) USNET.



(a) NSFNET (b) USNET

Fig. 10. Crosstalk per slot: (a) NSFNET, (b) USNET.

bandwidth are preferentially allocated to cores dedicated to that bandwidth, hence resource on each core is used efficiently. Nevertheless, MLFA-WT performs slightly better than On-demand in terms of spectrum utilization. With traffic prediction resource allocation is more effective, hence it is no surprise that the MLFA strategy performs significantly better than the other three. In particular, our MLFA algorithm improves spectrum utilization by about 35% (respectively, 30%) over the On-demand strategy for the NSFNET (respectively, USNET) topology at the highest traffic load in Fig. 9. The relative performance of the four strategies in Fig. 9 is a direct consequence of the relative performance of the same strategies in Fig. 8 in terms of fragmentation, and underscores the importance to network resource allocation of traffic prediction using machine learning techniques. The SU in NSFNET topology has similar tendency in USNET, since the resource in USNET is more than in NSFNET, the SU in USNET is slightly lower under the same traffic load.

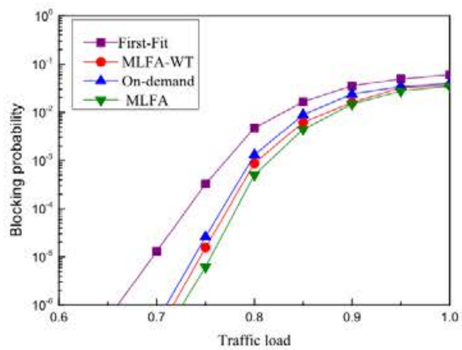
Fig. 10 shows the crosstalk per slot (CpS) metric as a function of traffic load for the four strategies and the two network topologies. First-Fit allocates spectrum on each core from lower to higher slot indices, hence adjacent cores will end up with many slots of the same index occupied by traffic, leading to serious crosstalk. This is evident in Fig. 10 as the crosstalk line for First-Fit lies well above the corresponding lines of the other three strategies across all traffic loads considered in our study. On the other hand, the MLFA-WT and MLFA strategies have almost the same performance in terms of crosstalk across all traffic load values and the two topologies. This result can be explained by observing that the two strategies use the same spectrum partitioning scheme for the various cores. Finally, the On-demand strategy employs a predefined core priority scheme [11] that is effective in reducing crosstalk. Nevertheless, our intra-core resource allocation strategy is seen to be more effective, especially at high loads where CpS under MLFA is about 22% (respectively, 18%) lower than the On-demand strategy for NSFNET (respectively, USNET).

Finally, Fig. 11 plots the blocking probability against the traffic load for the NSFNET and USNET topologies. The relative performance of the four strategies is identical to that shown in earlier figures and is a direct result of the effectiveness of each strategy to reduce/avoid fragmentation of the spectrum resources. Specifically, the First-Fit strategy exhibits the highest blocking probability, followed by the On-demand and MFLA-WT strategies that have similar performance. These two schemes reduce fragmentation and, therefore, may accommodate more connections. Our MFLA strategy performs even better; its blocking probability across the traffic load values simulated is on average about 40% (respectively, 28%) lower than that of the On-demand strategy for the NSFNET (respectively, USNET) topology. This improvement is due to the combination of two factors: (1) higher spectrum utilization, due to lower fragmentation, makes it possible to carry more traffic, and (2) lower crosstalk means that fewer connections are blocked due to physical layer impairments. The BP in USNET topology is lower, the reason is that USNET has higher node connectivity, where is easier to find alternative paths for the arriving requests.

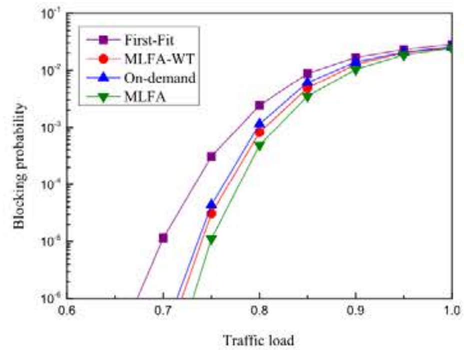
Overall, the results of our study presented in this spectrum demonstrate that traffic prediction using machine learning, combined with carefully designed intra- inter-core resource allocation strategies for MCF are quite effective improving network performance by alleviating the impact of both spectrum fragmentation and inter-core crosstalk.

7. Conclusion

SDM technology expands the capacity and flexibility of EONs, but it also introduces new challenges, namely, higher potential for fragmentation and inter-core crosstalk. In this paper, we presented a novel strategy that leverages machine learning to forecast future traffic



(a) NSFNET



(b) USNET

Fig. 11. Blocking probability.

demands to realize efficient resource allocation that reduces both crosstalk and fragmentation in SDM-EONs. We modeled the network resources as a 2D time-spectrum pool so as to simplify the process of spectrum allocation, and we introduced a spectrum partition scheme for adjacent cores of a link that can reduce inter-core crosstalk. The simulation results show that our strategy improves network performance in terms to crosstalk and blocking probability compared to earlier approaches.

Acknowledgments

This work was made possible with funding from: the National Natural Science Foundation of China (61401052); the Project of China Scholarship Council (201608500030); the Science and Technology Project of Chongqing Municipal Education Commission (KJ1400418, KJ1500445); the Starting Foundation for Doctors of Chongqing University of Posts and Telecommunications (A2015-09); and the Program for Innovation Team Building at Institutions of Higher Education in Chongqing (CXTDX201601020).

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.yofte.2019.03.001.

References

[1] S. Talebi, F. Alam, I. Katib, et al., Spectrum management techniques for elastic optical networks: a survey, Opt. Switching Netw. 13 (July 2014) 34–48.
[2] G.M. Saridis, D. Alexandropoulos, G. Zervas, et al., Survey and evaluation of space division multiplexing: from technologies to optical networks, IEEE Commun. Surv. Tut. 17 (4) (Aug. 2015) 2136–2156.
[3] S. Li, W. Lu, X. Liu, et al., Fragmentation-aware service provisioning for advance reservation multicast in SD-EONs, Opt. Express 23 (20) (2015) 25804–25813.
[4] M. Zhang, C. You, H. Jiang, et al., Dynamic and adaptive bandwidth defragmentation in spectrum-sliced elastic optical networks with time-varying traffic, J. Lightwave Technol. 32 (5) (2014) 1014–1023.
[5] C. You, M. Zhang, Z. Zhu, Reduce spectrum defragmentation latency in EONs with effective parallelization of connection reconfigurations, Proceedings of Optical Fiber Communication Conference (OFC), San Francisco, CA, 2014, pp. 1–3.
[6] M. Zhang, W. Shi, L. Gong, et al., Bandwidth defragmentation in dynamic elastic optical networks with minimum traffic disruptions, Proceedings of IEEE International Conference on Communications (ICC), Budapest, 2013, pp. 3894–3898.
[7] R. Zhu, J.P. Jue, A. Yousefpour, et al., Multi-path fragmentation-aware advance reservation provisioning in elastic optical networks, Proceedings of IEEE Global Communications Conference (GLOBECOM), Washington, DC, 2016, pp. 1–6.
[8] Y. Yin, H. Zhang, M. Zhang, et al., Spectral and spatial 2D fragmentation-aware routing and spectrum assignment algorithms in elastic optical networks [invited], J. Opt. Commun. Netw. 5 (10) (2013) A100–A106.
[9] N. Wang, J.P. Jue, X. Wang, et al., Scheduling large data flows in elastic optical inter-datacenter networks, 2015 IEEE Global Communications Conference (GLOBECOM), San Diego, CA, 2015.
[10] Q. Yao, H. Yang, H. Xiao, et al., Crosstalk-aware routing, spectrum, and core assignment in space-division multiplexing optical networks with multicore fibers, Opt. Eng. 56 (6) (2017) 066104.
[11] M. Yang, Y. Zhang, Q. Wu, Routing, spectrum, and core assignment in SDM-EONs with MCF: node-Arc ILP/MILP methods and an efficient XT-aware heuristic algorithm, J. Opt. Commun. Netw. 10 (3) (2018) 195.
[12] M. Klinkowski, P. Lechowicz, K. Walkowiak, et al., Survey of resource allocation schemes and algorithms in spectrally-spatially flexible optical networking, Opt. Switching Netw. 27 (2018) 58–78.
[13] H. Tode, K. Murakami, S. Fujii, et al., On-demand spectrum and core allocation for reducing crosstalk in multicore fibers in elastic optical networks, J. Opt. Commun. Netw. 6 (12) (2014) 1059–1071.
[14] Y. Zhao, L. Hu, R. Zhu, et al., Crosstalk-aware spectrum defragmentation based on spectrum compactness in space division multiplexing enabled elastic optical networks with multicore fiber, IEEE Access 6 (2018) 15346–15355.
[15] H. Yuan, M. Furdek, A. Muhammad, et al., A Space-division multiplexing in data center networks: on multi-core fiber solutions and crosstalk-suppressed resource allocation, IEEE/OSA J. Opt. Commun. Netw. 10 (4) (2018) 272–288.
[16] H. Tode, Y. Hirota, Routing, spectrum, and core and/or mode assignment on space-division multiplexing optical networks [invited], J. Opt. Commun. Netw. 9 (1) (2017) A99–A113.
[17] J. Mata, I. de Miguel, R.J. Duran, et al., Chamania, Artificial intelligence (AI) methods in optical networks: a comprehensive survey, Opt. Switching Netw. 28 (Apr.) (2018) 43–57.
[18] A. Yu, H.Yang, W.Bai, et al. Leveraging deep learning to achieve efficient resource allocation with traffic evaluation in datacenter optical networks, in: Proceedings of Optical Fiber Communication Conference (OFC), San Diego, 2018, pp. W4I.2.
[19] Q. Yao, H. Yang, R. Zhu, Mode Core, et al., Spectrum assignment based on machine learning in space division multiplexing elastic optical networks, IEEE Access 6 (Mar.) (2018) 15898–15907.
[20] Y. Xiong, J. Shi, Y.Y. Yang, George N. Rouskas, et al., Lightpath management in SDN-based elastic optical networks with power consumption considerations, J. Lightwave Technol. 36 (9) (2018) 1650–1660.
[21] F.J. Lin, Y.C. Hung, S.Y. Chen, et al., FPGA-based computed force control system using Elman neural network for linear ultrasonic motor, IEEE Trans. Industr. Electron. 56 (4) (2009) 1238–1253.
[22] Caida. <http://www.caida.org/home/>.
[23] Y. Xiong, X. Fan, S. Liu, Fairness enhanced dynamic routing and spectrum allocation in elastic optical networks, IET Commun. 10 (9) (2016) 1012–1020.
[24] K. He, W. Huang, Y. Jin, An efficient deterministic heuristic for two-dimensional rectangular packing, Comput. Oper. Res. 39 (7) (2012) 1355–1363.
[25] S. Fujii, Y. Hirota, H. Tode, et al., On-demand routing and spectrum allocation for energy-efficient AoD nodes in SDM-EONs, IEEE/OSA J. Opt. Commun. Networking 9 (11) (2017) 960–973.
[26] R. Zhu, Y. Zhao, H. Yang, et al., Multi-dimensional resource assignment in spatial division multiplexing enabled elastic optical networks with multi-core fibers, Proceedings of 2016 15th International Conference on Optical Communications and Networks (ICOCN) Hangzhou, 2016, pp. 1–3.
[27] J. Wang, W. Fang, H. Niu, et al., Financial time series prediction using Elman recurrent random neural networks, Comput. Intell. Neurosci. 12 (Jan.) (2016) 1–14.