Hindawi
Complexity
Volume 2023, Article ID 4140594, 13 pages
https://doi.org/10.1155/2023/4140594

Research Article

Resource Allocation in Multicore Elastic Optical Networks: A Deep Reinforcement Learning Approach

Juan Pinto-Ríos,[1] **Felipe Calderón**,[1] **Ariel Leiva**,[1] **Gabriel Hermosilla**,[1] **Alejandra Beghelli**,[2] **Danilo Bórquez-Paredes**,[3] **Astrid Lozada**,[4] **Nicolás Jara**,[4] **Ricardo Olivares**,[4] and **Gabriel Saavedra**[5]

[1] *School of Electrical Engineering, Pontificia Universidad Católica de Valparaíso, Av. Brasil 2950, Valparaíso 2362804, Chile*
[2] *Optical Networks Group, Department of Electronic and Electrical Engineering, University College London, WC1E 7JE, London, UK*
[3] *Faculty of Engineering and Sciences, Universidad Adolfo Ibáñez, Av. Padre Hurtado 750, Viña Del Mar 2562, 340, Chile*
[4] *Department of Electronic Engineering, Universidad Técnica Federico Santa María, Av. España 1680, Valparaíso 2390123, Chile*
[5] *Electrical Engineering Department, Universidad de Concepción, Víctor Lamas 1290, Concepción 4070409, Chile*

Correspondence should be addressed to Juan Pinto-Ríos; juan.pinto.r@pucv.cl

Received 17 June 2022; Revised 16 December 2022; Accepted 27 January 2023; Published 2 March 2023

Academic Editor: Ning Cai

A deep reinforcement learning (DRL) approach is applied, for the first time, to solve the routing, modulation, spectrum, and core allocation (RMSCA) problem in dynamic multicore fiber elastic optical networks (MCF-EONs). To do so, a new environment was designed and implemented to emulate the operation of MCF-EONs - taking into account the modulation format-dependent reach and intercore crosstalk (XT) - and four DRL agents were trained to solve the RMSCA problem. The blocking performance of the trained agents was compared through simulation to 3 baselines RMSCA heuristics. Results obtained for the NSFNet and COST239 network topologies under different traffic loads show that the best-performing agent achieves, on average, up to a four-times decrease in blocking probability with respect to the best-performing baseline heuristic method.

1. Introduction

Due to the ever-growing number of users, devices, and networking applications, Internet traffic keeps on increasing, more than doubling every two years, to levels that will lead to an eventual capacity crunch of the current core optical networks [1, 2]. Big technological companies, such as Google, Meta, Amazon, Netflix, Apple, and Microsoft now account for more than half of Internet traffic, and the introduction of 5G is expected to accelerate the growth of emerging heavy app users consuming 1 terabyte per month [3].

Various solutions to deal with this constant traffic growth have been proposed, ranging from greater efficiency in using currently deployed optical resources to expanding the capacity of the optical transport network. Examples of the former and latter are Elastic Optical Networks (EONs) [4] and multicore optical fiber (MCF) [5], respectively. EONs [6] divide the spectrum into narrow slots called frequency slot units (FSU), usually of 12.5 GHz width [7]. In EON communication, each connection uses as many adjacent slots as needed, thereby improving the spectral usage efficiency [8]. Under dynamic operation, EONs [9] can establish and release connections on-demand. MCF extends the fiber capacity by adding multiple cores within the same cladding. Thus, the capacity of a single fiber is significantly increased given that each core can be considered as an extra optical medium [10].

One of the first cases for the support of elastic optical networks was given by data-intensive applications running on multidata center systems [11]. Later, the need for elastic optical networks was highlighted for applications such as cloud-based IoT services [12], cloud-fog computing [13] as well as critical support for the 5G communication infrastructure [14] and the applications associated with it, such as Ultra High Definition videos, Telemedicine, and Smart City/Industry/Factory/Home [15]. Similarly, MCF has been identified as a complement of elastic optical networks to deliver the high capacity required by current and future applications, as well as the driving force to provide cost-efficient solutions for high-capacity submarine cables [16, 17], a key infrastructure underpinning Internet. Currently, applications requiring the combination of MCF [18] with the efficient use of the spectrum offered by dynamic elastic optical networks [19] have also been identified in scenarios such as intradata center networks [20, 21]. Going beyond the current technological state, expected Multimedia 3D Services for 6G networks such as Tactile/Haptic Internet, Video Games/Streaming as a 3D Service, and Deep-Sea Sightseeing [15] will certainly require a network capacity that only will be provided by the combination of MCF and dynamic EONs, termed as dynamic MCF-EON from now on.

One of the main challenges of dynamic MCF-EONs is the design of efficient routing, modulation, spectrum, and core assignment (RMSCA) strategies for establishing optical connections with as low blocking probability as possible. Most RMSCA proposals use heuristic approaches that consider the impact of intercore crosstalk (intercore XT) on optical signal quality, as described in [19, 22–25]. Although rule-based heuristics are computationally simple, their performance depends on the ability of the designer to detect the best set of rules defining the heuristic behavior [26]. In recent years, it has been shown that in most cases, deep reinforcement learning (DRL) techniques applied to solve resource allocation problems in dynamic elastic optical networks outperform rule-based systems [27, 28]. DRL has the ability to explore solutions other than those detected by the expert knowledge of the human designer. As a result, it has the potential of generating new nonobvious policies from the experience gained after training in a relevant environment [29].

1.1. Related Work. In dynamic scenarios, DRL was applied to solve the routing, modulation, and spectrum assignment (RMSA) problem in single-domain EONs [27, 28, 30, 31], multidomain EONs [32], multiband EONs [33, 34] ,and survivable EONs operating under shared protection [35]; the problem of energy-efficient traffic grooming in fog-cloud EONs [36], the problem of establishing and reconfiguring multicast sessions in EONs [37], the fragmentation mitigation problem [38], and the resource allocation problem with advanced reservation (AR) in EONs for cloud-edge computing [39]. Only one previous work has studied the application of DRL on MCF networks [40], but this work focused on fixed-grid networks. In this paper, we extend the work reported in [27, 30, 31] by applying DRL to dynamic MCF-EONs for the first time.

In the context of dynamic MCF or MCF-EON networks, with the exception of [40], only supervised machine-learning techniques have been applied so far. These consist of techniques for making inferences based on expert-labeled data. Thus, instead of taking actions, supervised learning algorithms perform estimations or classifications [41]. For example, the authors of [42, 43] used supervised learning to predict future connection requests in dynamic MCF-EONs to perform a crosstalk-aware resource allocation in advance. Instead, the authors in [44] used machine learning to estimate the intercore XT to then execute a crosstalk-aware allocation algorithm. All these studies have used machine learning as an auxiliary process to improve the heuristic allocation, either by predicting future traffic or transmission quality. In none of them, machine learning had direct participation in the decision-making related to resource allocation.

1.2. Paper Contribution. To the best of our knowledge, there are no previous studies on applying DRL to solve the RMSCA problem in dynamic MCF-EONs. In this paper, we present, for the first time, the implementation and testing of a new dynamic MCF-EON environment where four different DRL agents are trained to solve the RMSCA problem. The results obtained by the best-performing agent are then compared to 3 baseline heuristics.

The rest of this article is organized as follows: Section 2 presents the DRL system developed, Section 3 describes the performance evaluation experiments, and Section 4 concludes the paper.

2. DRL for Dynamic MCF-EONs

A DRL system can be summarized as an agent (an entity equipped with a learning algorithm) that—during its training phase—learns to make good decisions by interacting with an environment [45, 46].

In the context of RMSCA, the agent must learn to allocate optical resources to connection requests such that they are not blocked. Blocking can happen due to physical impairments or lack of spectral continuity or contiguity in the chosen route. A good allocation decision makes the environment give the agent a high-value reward.

Formally, a DRL system can be modeled as a Markov Decision Process (MDP) described by the 6-tuple $\{\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, s_0, \gamma\}$ [29], where the following takes place:

(i) $\mathcal{S}$ (States): Set of possible states describing the status of the system. In this work, the state s_t is described by the link spectrum utilization, at time step t, of each candidate route per core between the source and destination node of connection request cr_t. The latter is defined by the tuple $\{o, d, h, b\}$, where o is the source node, d is the destination node, h is the holding time of the request, and b is the bitrate of the demand.

(ii) $\mathcal{A}$ (Actions): Set of actions the agent can take. In this work, an action a_t at time step t is a triplet (k, c, j), where k is the selected route (out of K pre-computed routes) c the identifier of the core (out of C cores), and j the identifier of a block of contiguous slots that can accommodate the demand of cr_t (out of J blocks).

(iii) $\mathcal{T}(s_{t+1}|s_t, a_t)$ (Transition probability): Probability distribution that the system transits to state s_{t+1}, given the system is in state s_t and the agent takes action a_t when receiving connection request cr_t.

(iv) $\mathcal{R}(s_t, a_t, s_{t+1})$ (Reward): The reward function that defines the immediate reward (r_t) received when transiting to state s_{t+1} due to action a_t while in state s_t. In this work, the reward is designed to be equal to 1 if the request is accepted and −1 if it is rejected.

(v) s_0 (Initial state): The state of the network at the start of the decision process. In this work, this state corresponds to all routes having all spectrum slots available in all links and cores.

(vi) γ (Discount factor): A parameter $\in [0, 1)$ that sets the importance of current and future rewards. This factor adjusts the process of exploration and exploitation of agents in the environment [29].

The evolution of the DRL system defined above is as follows: During a training episode—made of a finite amount of time steps—an agent learns to make good decisions by interacting with the environment at each time step t [45, 46]. To do so, upon receiving a connection request cr_t with the system in state s_t, the agent generates an action a_t. Such action makes the environment transit to the state s_{t+1} with probability $\mathcal{T}(s_{t+1}|s_t, a_t)$ and the agent receives a reward $R_t(s_t, a_t, s_{t+1})$. The objective of the agent is maximizing the expected future discounted reward. Thus, by repeating this process during the training episode, the agent will learn a policy $\pi^*(a|s)$ that leads to maximizing the return function, Γ_t, defined as

$$\Gamma_t = \sum\nolimits_{t\in[t,\infty)} \gamma^{t'-t} \cdot R_{t'}. \tag{1}$$

The details of the state modeling are as follows: We extend the state defined in [27] by considering the different cores. Thus, the state is represented as an array of $1 \times (2|V| + 1 + (2J + 3) \cdot K \cdot C)$ elements, where $|V|$ is the number of nodes of the optical network. The extended state is then given by

$$\begin{aligned} s_t = \Big\{ o, d, h, b, \Big\{ \big\{ z_{k,c}^{1,j}, z_{k,c}^{2,j} \big\} \Big|_{j\in\{1,\ldots,J\}} \Big\}, \\ z_{k,c}^{3}, z_{k,c}^{4}, z_{k,c}^{5} \Big\}_{k\in\{1,\ldots,K\}} \Big|_{c\in\{1,\ldots,C\}}, \end{aligned} \tag{2}$$

where a one-hot encoding is used to identify the origin and destination nodes. Each subcomponent z in the vector mentioned above is as follows: For each core $c \in C$ and route $k \in K$ between o and d nodes, $z_{k,c}^{1,j}$ is the size of $j - th$ block that can accommodate the connection request. $z_{k,c}^{2,j}$ is the index of the first slot of each block j. The third component, $z_{k,c}^{3}$, is the number of FSUs required to establish the connection (given the modulation format used). Finally, $z_{k,c}^{4}$, the average number of FSUs available in all J blocks in route k and core c is included, and $z_{k,c}^{5}$ is the total number of FSUs available in the route k in core c.

In our resource allocation problem, the environment is programmed to represent the operation and constraints of a dynamic MCF-EON. When a connection request arrives during the training phase, the agent decides what resources to allocate. At the beginning of its training, the agent makes random decisions (exploration process). Then, the environment determines whether the set of resources identified by the agent is feasible and gives the agent feedback about the quality of its decision. This information, stored in the experience buffer of the agent, allows the agent to learn. As a result, it starts to select better actions (exploitation process) for future requests. Better actions result in the agent earning a high cumulative reward. After an agent has finished its training stage, it can be evaluated (testing stage) by having it to process a new set of connection requests.

The implementation of any DRL system is done in two stages as follows:

(i) *Stage 1: Environment Design and Implementation.* The environment is a program that receives the agent's action, processes it, and sends back feedback. The specific feedback depends on the results of the agent's action on the environment. The environment must consider the characteristics and constraints of the existing system to process the action. In the case of an optical network, the environment must manage information about the network topology and status and model the network operation (including physical phenomena related to the signal transmission and spectrum allocation constraints).

(ii) *Stage 2: Agent Training.* The agent must first acquire knowledge about the environment. This training is done by exploration and exploitation. When exploring, the agent selects random actions to learn how the environment reacts and stores such knowledge. When exploiting stored knowledge, the agent makes informed decisions to select the following action: During exploration and exploitation, the agent receives feedback from the environment, which the agent uses to update its knowledge (policy). In this way, the agent's training progresses.

In the following section, these two stages are described in detail in the context of dynamic MCF-EONs.

2.1. Stage 1: Environment Design and Implementation.

In this work, the toolkit *Optical RL-Gym*, developed by Natalino and Monti [31] to facilitate the implementation and replicability of deep reinforcement learning environments for optical networks was extended by creating a new environment: DeepRMSCAEnv. Such an environment encapsulates all the necessary functions to simulate an MCF-EON.

The right part of Figure 1 shows a schematic of the implemented environment, including its main components and interactions. Dashed and thick lines modules are modules from the Optical RL-Gym toolkit that had to be modified and developed from scratch, respectively, to model an MCF-EON environment correctly.

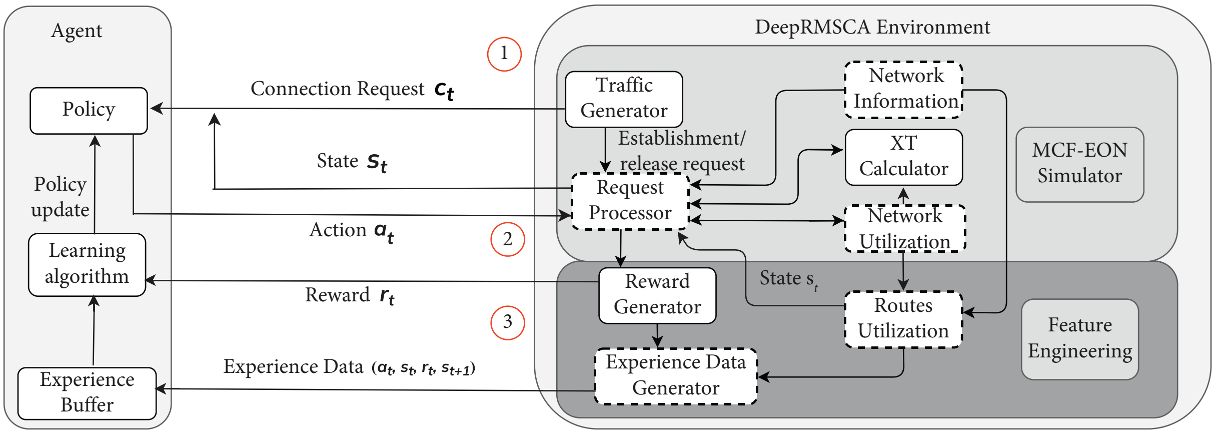

Figure 1: Interaction between a DRL agent and the MCF-EON environment developed: DeepRMSCAEnv.

The environment can be considered made of an event-driven dynamic MCF-EON simulator and a feature engineering module. The former is responsible for processing the connection requests according to the agent's action and sending the relevant information to the feature engineering module. The latter is responsible for preparing and sending feedback to the agent (reward and observation).

The dynamic MCF-EON simulator consists of five components. Two of these store data about the network as follows:

(i) *Network Information*. This component stores the graph representation of the network and the link capacity, considering the multicore nature of links. It also stores the K alternatives routes for each source-destination pair, the modulation format used as a function of the route length distance, and the network information, coded as in [27].

(ii) *Network Utilization*. This component stores the utilization of each slot (available or used) for each network link and core.

The remaining 3 components perform specific tasks

(i) *Traffic Generator*. This component is responsible for the random generation of connection establishment and release requests. At time step t, connection request cr_t is sent to the agent and the request processor component. Connection release requests are sent only to the request processor.

(ii) Request Processor. This component receives several inputs. The first two are the connection establishment or release request and the action of the agent (in the case of a connection request). When connection requests cr_t are received at time step t, the Request Processor module receives s_t from the Route Utilization module and sends it to the agent, then the Request Processor module waits for the action of the agent. Once the action, a_t, is received, the Request Processor first determines the number of slots the connection requires. To do so, the most efficient modulation format that ensures a QoT [47] is first selected (QoT has been transformed into a maximum reach, as shown in Table 1). The calculation of the number of slots is the same described in Section 2 of [27]. Next, it checks the network topology (input from the network information module) and the network utilization (input from the network utilization module) to evaluate the availability of the resources selected by the agent. It also obtains information from the XT calculator component regarding the feasibility of the allocation in terms of crosstalk. If resources are available and a positive answer is received from the XT calculator, then resources are allocated, and the corresponding information is updated on the network utilization module. Information about a successful establishment is also sent to the Reward Generator module. If resources cannot be allocated, information about the failed establishment is sent to the Reward Generator component only. When a connection release is received, the Request Processor component updates the network utilization module to make the released resources available.

(iii) *XT Calculator*. This component calculates the intercore crosstalk (XT), defined as the interference between optical connections in neighboring cores using the same frequency slots. It receives information about the resources selected by the agent's action a_t (length of the links composing the route and core) and the route-level utilization information from the network utilization module and evaluates the XT. For generic MCF systems, with any number of cores in any geometric arrangement, the steps to calculate the mean XT affecting a connection established in core x are as follows:

(a) Calculate the mean XT per unit of length between core x and adjacent core y, $w_{x,y}$ as

$$w_{x,y} = \frac{2g^2 q}{\beta \Lambda_{x,y}}, \tag{3}$$

where g, q, β, and Λ are the coupling coefficient, radius of curvature (or bending), constant propagation, and the distance between cores x and y, respectively.

(b) Calculate the total mean XT affecting core x, XT_x, by adding the crosstalk contribution of all its adjacent cores. That is,

$$\mathrm{XT}_x = \sum_{y=1}^{n} w_{x,y} \cdot L, \tag{4}$$

where n is the number of cores adjacent to core x and L the length of the link.

For the specific case where cores follow a triangular or hexagonal geometric arrangement and different pairs of cores are equidistant, equation (5) has been found to be a better approximation to calculate XT_x [19], as given as follows:

$$\mathrm{XT}_x = \frac{n - n \cdot \exp\left[-(n+1) \cdot wL\right]}{1 + n \cdot \exp\left[-(n+1) \cdot wL\right]}, \tag{5}$$

where, as in equation (4), n represents the number of cores neighbouring x, and L is the length of the link. The term w is given by equation (3) (subindices have been dropped since the distance between all core pairs is assumed to be the same).

An XT threshold value for different modulation formats is defined in [48, 49] such that the signal quality is acceptable. If XT exceeds this predefined threshold (summarized in Tables 1 and 2), a negative answer is sent to the request processor (−1). Otherwise, a positive answer is sent (1).

The feature engineering module in Figure 1 is responsible for preparing the information to be sent back to the agent. It is made of three components as follows:

(i) *Reward Generator.* This component calculates the numerical reward to be sent to the agent depending on the information received from the Request Processor component. In this work, a successful resource allocation returns a reward equal to 1 and a failed allocation equal to −1. Connections can be rejected due to a lack of spectrum resources along the route selected by the agent, because of crosstalk among cores exceeding the predefined threshold, or because the length of the route selected by the agent is longer than the maximum optical reach of any modulation format (such limit depends on the modulation format and a bit-error-rate threshold, as in Table 1 of [50].

(ii) *Routes Utilization.* This component receives the routing information from the network information component and the utilization state of the slots in the K shortest routes between the origin and destination nodes of connection request cr_t from the network utilization module. This information is then consolidated in a 1D vector made of $(K \cdot C \cdot J)$ elements, where K is the number of alternative routes, C is the number of cores, and J is the number of blocks with enough available slots to establish the connection request being processed.

(iii) *Experience Data Generator.* This component builds the information to be stored in the Experience Buffer which is a collection of tuples (a_t, s_t, r_t, s_{t+1}) generated during the training process.

Table 1: Maximum reach for each modulation format [50].

Modulation format	Max. reach (km)
64QAM	250
32QAM	500
16QAM	1000
8QAM	2000
QPSK	4000
BPSK	8000

Table 2: XT threshold for each modulation format [49].

Modulation format	XT threshold (dB)
64QAM	−34
32QAM	−27
16QAM	−25
8QAM	−21
QPSK	−18
BPSK	−14

2.2. Stage 2: Agent Training.

2.2. Stage 2: Agent Training. The left side of Figure 1 shows the interaction between the agent and the DeepRMSCAEnv environment during the training stage.

The agent aims to maximize its long-term reward. That is, selecting actions leads to the highest number of connection requests established. To achieve this goal, the agent is built considering two main components.

2.2.1. Policy. This component is where the knowledge of the behavior of the agent is embedded. At a given time, t receives a connection establishment request, cr_t as input along with state s_t, and action a_t is outputted. The action is defined by 3 integer numbers, namely, a route identifier k (selected out of K possible precomputed routes), a core identifier c (selected out of C possible cores), and the identifier j of the block selected. These values define which route, core, and spectrum resources should be assigned to each request. As the agent successfully allocates more connection requests, the policy becomes better. At the end of the training, the policy is expected to allow the agent to define which action has the highest probability of not being blocked. Figure 2 shows a simplified example of two possible actions that might be taken by the agent, given a specific s_t.

On the left part of the figure, a 5-node network topology and a connection establishment request of 2 slots between nodes 5 and 3 are shown. The demand is represented by the

Table 3: Network, traffic and training parameters.

Parameters	Value
Network parameters	
Topologies	NSFNet [50] and COST239 [51]
Number of cores	3
Number of FSU by link	100
Modulation formats	BPSK, QPSK, 8-QAM, 16-QAM
Traffic parameters	
Bit rates (Gb/s)	Uniformly distributed in [25–100] Gbps
Agent training parameters	
Precomputed candidate routes	5
Number of connection requests per episode	50 [27]
Simulated requests per training	160,000
Agent's learning algorithm parameters	By-default [52]
Agent's hyperparameters	By-default [31]

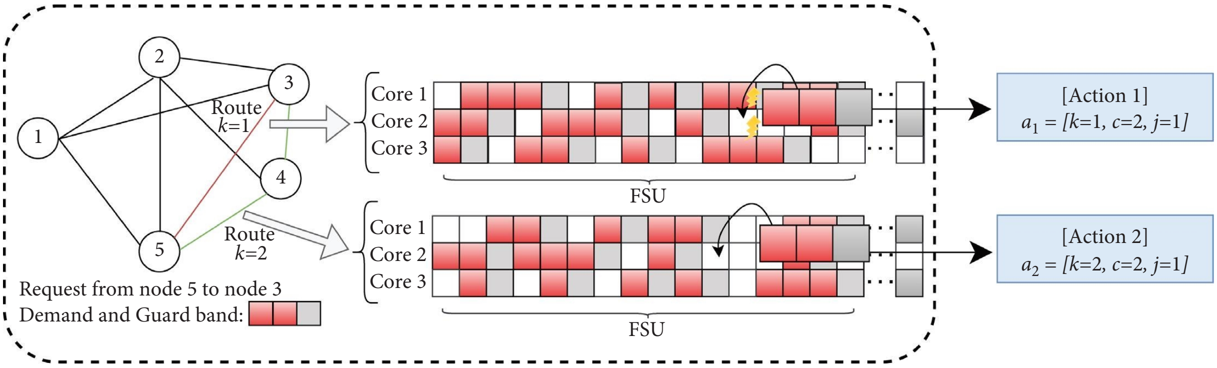

Figure 2: Example of a connection request from node 5 to 3, requesting 3 slots (2 for data, 1 as guard band). $K = 2$, meaning 2 routes. The routes spectral use is represented by white blocks (available FSUs) and red and grey ones (occupied for data and as guard bands, respectively).

red boxes (2 slots in this case) plus the grey box (1 slot used as a guard band). The number of slots required to serve the connection (red squares) is determined by the modulation format, using the same method presented in [27]. One guard band of 1 slot is considered for each connection request to achieve a good trade-off between the quality of transmission and the blocking probability [51].

Let us assume that the network is equipped with three cores per link, and the agent can select either route 1 ($k = 1$), represented by the red link in the topology, or route 2 ($k = 2$) by the green links. In addition to a route, the agent must also select a core and a slot. On the right side of the figure, the spectrum utilization of both routes is shown. Red and grey squares represent used FSUs. A row of squares represents the slot utilization in a specific core for a specific route. Thus, the three rows on the upper and lower part of the figure represent the slot utilization on the three cores of the first and second routes, respectively.

If the agent selects Action 1, depicted in the upper part of the figure, then action $a_t = [1, 2, 1]$ is sent back to the environment, signaling that the agent selects slot 11 as the initial slot on route 1 in core 2 to establish the connection. The thunderbolt symbol in route 1 represents the presence of crosstalk exceeding the acceptable threshold. In this case, the request will be rejected, and a reward of −1 will be sent to the agent. Instead, if the agent selects Action 2, depicted on the lower part of the figure, then action $a_t = [2, 2, 1]$ is sent back to the environment. This action leads to a successful connection establishment, and the agent receives a reward equal to 1. During the training stage, the policy component should be updated to select Action 2 over Action 1 (for this state s_t), leading to a higher reward.

2.2.2. Learning Algorithm. This component receives the Experience Data from the environment and, based on that information, updates the policy to produce actions that maximize the expected cumulative long-term reward. In this study, we consider learning algorithms compatible with the action space. The action space used has a multidiscrete nature because the action is defined by multiple discrete values (route, core, and slot identifier). Thus, the learning algorithms available in the Stable-Baselines [52] library that was compatible with a multidiscrete space state were selected (as also done in [27, 31]). These are as follows:

(i) *Advantage Actor-Critic (A2C)* [53] *and Actor-Critic using Kronecker-Factored Trust Region (ACKTR)* [54]. These are approaches based on the actor-critic algorithm [53], which has two interacting neural networks. The actor uses a dense neural network to process and update the policy obtained. The critic uses a separated neural network to evaluate the quality of the policy by calculating the "value function" [45]. Both algorithms differ in how they update their neural networks' weights. A2C does that by using the feedback the critic's network gives to the actor's network, whilst ACKTR uses a Kronecker-factored approximation [56], which is a method that optimizes the stochastic gradient descent.

(ii) *Proximal Policy Optimization (PPO2)* [55] *and Trust Region Policy Optimization (TRPO)* [56]. These learning algorithms use only one neural network, whose weights are updated based on the policy gradient descent. They differ in the way the policy gradient descent is approached. TRPO avoids sudden changes in the neural network weights, updating only those that do not differ by a greater distance than what the Kullback–Leibler restriction (relative entropy) [56] allows. Instead, PPO2 does not impose limits on the neural network weights' changes to optimize the policy's descent curve.

3. Performance Evaluation

Table 3 lists the values of the main parameters used to train the agents. In terms of network parameters, we consider two topologies, namely, the NSFNet (mills [57]) and the COST239 (Batchelor [58]). For each one, we assume 100 FSUs and 3 cores arranged in a triangular geometry per link, and the available modulation formats are BPSK, QPSK, 8-QAM, and 16-QAM. These simplifications have been considered due to memory constraints. The same number of slots was considered in [33]. As in [59], we use (5) to calculate the XT.

Regarding the traffic characteristics, we assume a fully dynamic behavior, where connection establishment requests arrive as a Poisson process and connection holding times follow a negative exponential distribution. The bitrate associated to each connection is uniformly selected from the range [25–100] Gbps, as in [27]. Finally, regarding the agents (one per learning algorithm), they will select one out of 5 precomputed routes, one out of 3 cores, and the identifier j of the FSU block for the connection considering a total of 100 FSUs. Agents will be trained in episodes made of 50 connection requests each (to simplify backpropagation in the dense neural network used by the agent by delivering small batches of data continuously), and the whole training session will consider a total of 160,000 connection requests. The parameters of the four agents will be the ones set by default in the agent's library *Stable Baselines* [52]. The DRL system developed is available in a Git repository (The new environment, under the name DeepRMSCAEnv, is available at https://gitlab.com/IRO-Team/deeprmsca-a-mcf-eon-enviroment-for-optical-rl-gym/).

3.1. Preliminary Training Results. Results for the training process of 4 agents are presented. The following discussion about the results obtained is valid only for the hyperparameters used for each agent defined in Table 3.

The agents TRPO, PPO2, A2C, and ACKTR were trained with a traffic load of 250 Erlang, as in [27].

Figures 3 and 4 show the reward accumulated by the different agents during their training in the NSFNet and COST239 topologies, respectively. Given that each episode is made of 50 connection requests, the maximum reward achievable by an agent is 50. It can be seen that the A2C and TRPO agents are the only ones reaching values close to the maximum expected reward in both topologies with an average reward of 49 and 47, respectively, with TRPO exhibiting slightly better performance. On the other hand, the PPO2 and ACKTR agents did not perform well using the default parameters. PPO2 performed well on the COST239 topology (reward oscillated around 42) but not on NSFNet (reward oscillated around 30). In both topologies, the agent got stuck to the same value of reward from the very beginning, showing no signs of learning. In the case of ACKTR, the default parameters were not suitable for this task either. Not only the agent got low values of reward in both topologies, but in the COST239 topology, the reward obtained decreased during several periods of the training process, never again exceeding the value obtained in the first 10,000 timesteps.

The A2C (Actor-Critic) learning algorithm filters those agents' actions leading to a low reward. Such filtering is possible thanks not only to the feedback received from the environment but also to the feedback given to the Actor (neural network in charge of applying the policy) by the Critic (neural network in charge of evaluating the quality of the policy used through the Value function). As a result, the agent starts with low values of reward (exploration phase) to then quickly increasing its reward per episode (exploitation phase) as the training progresses. Such behavior can be observed in Figures 3 and 4, where the A2C agent requires a few episodes to achieve a reward close to 50 and exhibits one of the best results in both topologies.

ACKTR (Actor-Critic using Kronecker-Factored Trust Region) is a trust-region optimization algorithm for actor-critic methods with gradient update sped up by means of the Kronecker-factored approximation. The effectiveness of the trust-region method is highly dependent on the learning algorithm's parameters. In practice, using the by-default parameters of the Stable Baselines led to the following: (a) the weights of the actor's neural network not being updated, trapping the agent in a local optimum, as seen in Figure 3 (NFSNet topology) and (b) not finding the trust region, resulting in random actions, as seen in Figure 4 (COST 239 topology).

PPO2 uses a different approach by updating the gradient more frequently than other methods. As a result, it can find a good policy more quickly than other methods, as shown in Figures 3 and 4. However, Figure 3 shows that it also gets

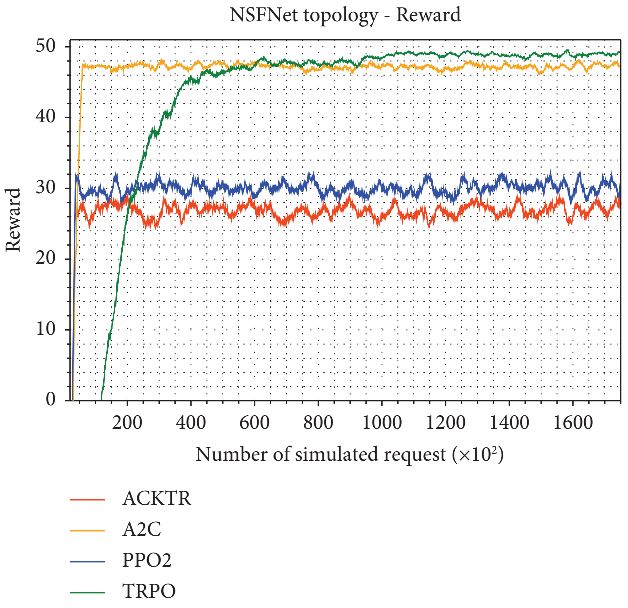

Figure 3: Accumulated reward for the A2C, PPO2, TRPO, and ACKTR agents in the NSFNet topology.

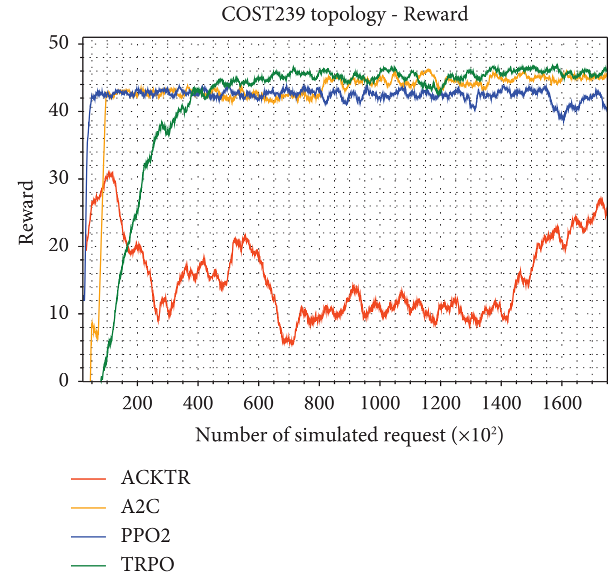

Figure 4: Accumulated reward for the A2C, PPO2, TRPO, and ACKTR agents in the COST239 topology.

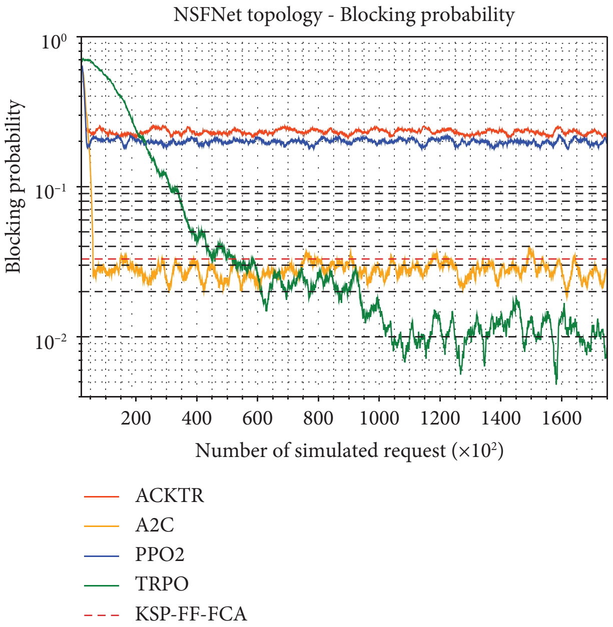

Figure 5: Blocking probability for A2C, PPO2, TRPO, and ACKTR in NSFNet Topology.

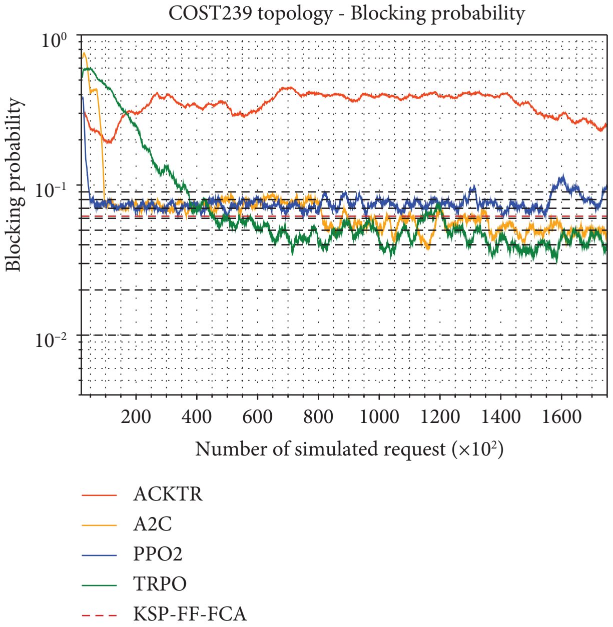

Figure 6: Blocking probability for A2C, PPO2, TRPO, and ACKTR in COST239 Topology.

stuck in a local optimum. Most probably this is due to the use of the by-default learning algorithm's parameters of Stable Baselines.

Finally, TRPO combines the policy gradient method of PPO2, but it also uses a trust region to avoid radical changes in the update of the neural network weights. The size of the trust region is aimed to avoid increasing the relative entropy of information based on the factor Kullback–Lieber As a result, it improves slowly and monotonically, as seen in Figures 3 and 4. For the problem studied here, this agent achieved the highest cumulative reward.

Please notice that parameter tuning is out of the scope of this work, as our aim was to show the potential of DRL as a solution for the dynamic resource allocation in MCF-EONs.

Figures 5 and 6 show the evolution of the blocking probability during the training process of the same agents for the NFSNet and COST239 topologies, respectively. For comparison, the dashed red line shows the blocking probability obtained by one of the baseline heuristics, kSP-FF-FCA. This heuristic has a list of 5 precomputed routes, sorted from shortest ($k = 1$) to longest ($k = 5$). When a connection request arrives, the heuristic attempts to establish the connection in the shortest path of the list ($k = 1$), applying the first-fit policy for spectrum allocation and first-fit crosstalk-aware for core allocation, as described in [60]. The same procedure is repeated for the following route in the list if unsuccessful: After attempting all paths, the connection is rejected if there are no available resources.

From the figure, we can see that once the agents are in steady-state, TRPO and A2C agents outperform the heuristic, improving blocking of 24.3% and 73.9% for the NSFNet topology and 14.51% and 38.71% for the COST239 topology, respectively.

Given the excellent performance of the TRPO agent in both topologies, in the following section, this agent will be trained for different traffic loads, and then its performance will be contrasted with that of the heuristics selected in [61].

3.2. TRPO Training Results.

The TRPO agent was trained for traffic loads between 500 and 3000 Erlang, in steps of 500. Figures 7 and 8 show the evolution of the blocking probability achieved by the TRPO agent as the training process progresses for different traffic loads for the NSFNet and COST239 topologies, respectively. It can be seen that the agent exhibits consistent behavior, with the blocking probability increasing with the traffic load, as expected. It can also be seen that at the beginning of the training process, the agent obtains a high blocking probability due to the exploration process. When the exploitation process starts, the blocking probability is reduced until it converges to a steady value. This happens after 150 thousand timesteps for the NFSNet and 130 thousand timesteps for COST239, irrespective of the traffic load. Given this steady value, we assume training has finished and the trained agent can now be evaluated in a testing setting.

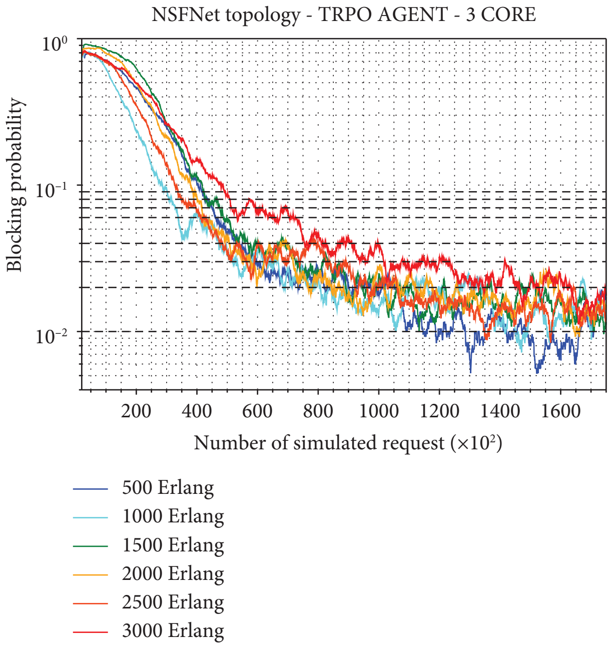

FIGURE 7: Blocking probability progress for TRPO agent training in NSFNet Topology.

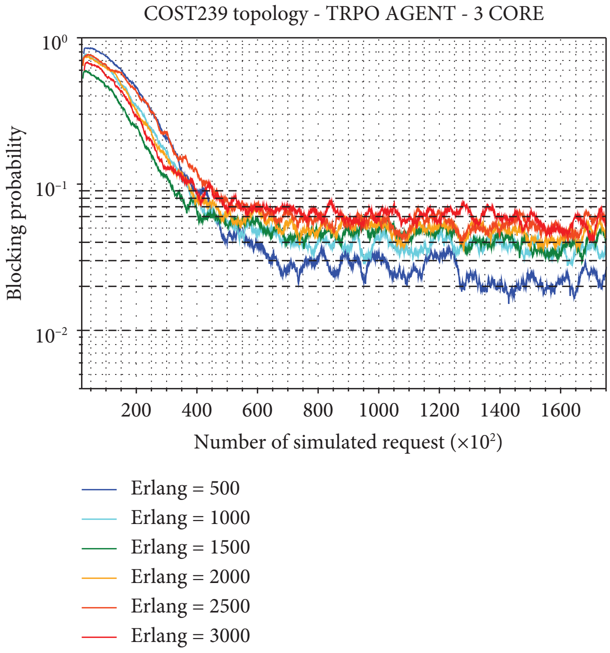

FIGURE 8: Blocking probability progress for TRPO agent training in COST239 Topology.

3.3. TRPO Agent VS. Heuristic: Blocking Performance.

Figures 9 and 10 show the blocking probability achieved by the trained TRPO agent and the same heuristics, selected for blocking evaluation in the survey [50], namely, KSP-FF-FCA [61], KSP-RF-RCA [61], and KSP-SCMA XT/demand-aware [22]. Results assume operation in the C-band (320 FSU) for the NSFNet and COST239 topologies, respectively. The three heuristics apply alternated routing. KSP-FF-FCA uses the First Fit policy to select core and spectrum, KSP-RF-RCA applies a random policy to select core and spectrum, and KSP-SCMA XT/demand aware allocates different parts of the spectrum and core depending on the bitrate of the connection request. If the connection request's demand is below a bitrate's threshold, a First-Fit allocation policy is applied for spectrum and core assignment as long as the cross-talk levels are not exceeded; otherwise, a Last-Fit policy is applied if the connection request's demand is above the threshold.

Compared to the best-performing heuristic, KSP-SCMA XT/demand-aware, a significant improvement in the blocking performance of the DRL approaches is observed. For example, in the NFSNet topology, at the highest load studied, the TRPO agent exhibits a blocking probability of about $1.9 \cdot 10^{-2}$, about four times slower than the blocking of $8.5 \cdot 10^{-2}$ achieved by the heuristic. On average, considering both topologies and loads over 2000 Erlang, TRPO achieves a 4-times decrease in blocking concerning the best heuristic, being ideal for the future scenario of demand for connection requests [64] and highlighting the benefits of applying DRL techniques to the RMSCA problem.

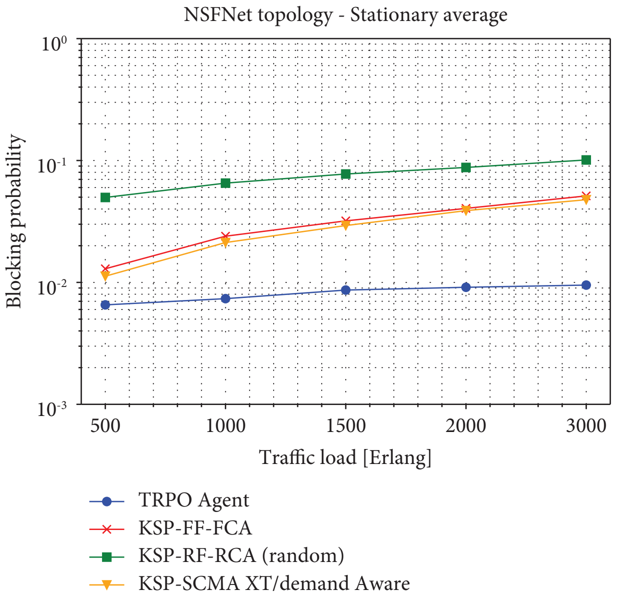

Figure 9: Blocking probability steady average of TRPO agent trained in NSFNet topology.

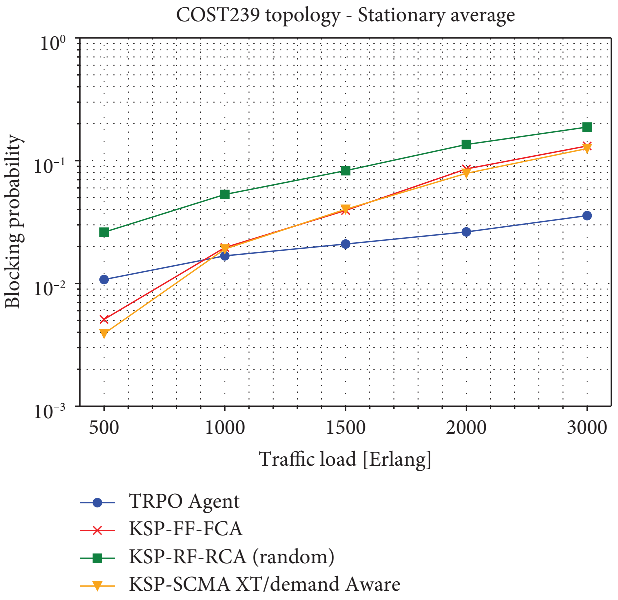

Figure 10: Blocking probability steady average of TRPO agent trained in COST239 topology.

Finally, our results show that the trained agent can generalise policies for different traffic loads and spectrum resources and outperform the rule-based heuristics. The improved performance comes from the ability of the DRL to explore solutions other than those detected by the expert knowledge of the human designer of the heuristics. We have also observed that training in adverse conditions achieves good results. That is, training the agent at high traffic loads makes the agent to perform well at lower traffic loads whereas training the agent using links with reduced capacity leads to the agent to perform better in links with increased capacity. In line with previous research [33], such generalisation was not observed in terms of topology: The agent trained in the NFSNet topology did not perform well in the COST 239 topology and vice versa. Studying the benefit of using Graph Neural Networks to overcome the lack of topology generalisation is part of current research [63].

4. Conclusion

This paper presents a deep reinforcement learning approach applied for the first time in the literature to solve the routing, modulation format, spectrum, and core allocation problem in dynamic multicore elastic optical networks. Simulation results show that the deep reinforcement learning approach offers a significant performance advantage over the best heuristic strategy studied.

Further research on improving the DRL approach performance should focus on hyperparameter tuning, applying transfer learning techniques or graph neural networks to cover a broader range of topologies with decreased computational effort, increasing the size of the data to be processed to study fibers with more cores and investigating different reward schemes that differentiate the reward according to the cause of blocking (e.g. crosstalk, capacity unavailability, fragmentation, or optical reach).

Additionally, we would like to explore explainability techniques that might help understand how the agent makes its decisions to improve current heuristics.

We expect these results and the code made available in the Git repository to help the research community study the benefits of deep reinforcement learning in the area of optical networks.

Data Availability

The data used to support the findings of this study are available at https://gitlab.com/IRO-Team/deeprmsca-a-mcf-eon-enviroment-for-optical-rl-gym.

Disclosure

The preprint is available in Arxiv at https://doi.org/10.48550/arXiv.2207.02074 and in DeepAI blog at: https://deepai.org/publication/resource-allocation-in-multicore-elastic-optical-networks-a-deep-reinforcement-learning-approach.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

Financial support from DI-PUCV (039.437/2020, 039.382/2021); ANID FOVI 210082; ANID FONDECYT Iniciación (11201024, 11220650, 11190710); ANID Magister Nacional (/2020-22201418, /2021-22210736); ANID Doctorado Nacional/2022-21220867 is gratefully acknowledged.

References

[1] C. W. Paper, “Cisco visual networking index: global mobile data traffic forecast update,” 2018, http://media.mediapost.com/uploads/CiscoForecast.pdf.

[2] TeleGeography, “State of the networks,” 2022, https://www2.telegeography.com/hubfs/LP-Assets/Ebooks/state-of-the-network-2022.pdf.

[3] A. Weissberger, “Sandvine: Google, facebook, microsoft, apple, amazon & netflix generate almost 57% of internet traffic,” 2021, https://tinyurl.com/2uerpfv2.

[4] M. Jinno, H. Takara, B. Kozicki, Y. Tsukishima, Y. Sone, and S. Matsuoka, “Spectrum-efficient and scalable elastic optical path network: architecture, benefits, and enabling technologies,” *IEEE Communications Magazine*, vol. 47, no. 11, pp. 66–73, 2009.

[5] T. Mizuno, H. Takara, A. Sano, and Y. Miyamoto, “Dense space-division multiplexed transmission systems using multicore and multi-mode fiber,” *Journal of Lightwave Technology*, vol. 34, no. 2, pp. 582–592, 2016.

[6] Y. Ujjwal and J. Thangaraj, “Review and analysis of elastic optical network and sliceable bandwidth variable transponder architecture,” *Optical Engineering*, vol. 57, no. 11, pp. 1–18, 2018.

[7] R. Zhou, M. D. Gutierrez Pascual, P. M. Anandarajah, T. Shao, F. Smyth, and L. P. Barry, “Flexible wavelength de-multiplexer for elastic optical networking,” *Optics Letters*, vol. 41, no. 10, pp. 2241–2244, 2016.

[8] J. Wu, S. Subramaniam, and H. Hasegawa, “Comparison of oxc node architectures for wdm and flex-grid optical networks,” in *Proceedings of the 2015 24th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–8, Las Vegas, NV, USA, August 2015.

[9] C. Politi, T. Orphanoudakis, E. Kosmatos, and H. C. Leligou, “Dynamic resource allocation in elastic optical networks,” in *Proceedings of the 2015 17th International Conference on Transparent Optical Networks (ICTON)*, pp. 1–4, Budapest, Hungary, July 2015.

[10] Y. Awaji, K. Saitoh, and S. Matsuo, “Chapter 13 - transmission systems using multicore fibers,” in *Optical Fiber Telecommunications*, I. P. Kaminow, T. Li, and A. E. Willner, Eds., pp. 617–651, Academic Press, Boston, MA, USA, 6th edition, 2013.

[11] P. Lu, L. Zhang, X. Liu, J. Yao, and Z. Zhu, “Highly efficient data migration and backup for big data applications in elastic optical inter-data-center networks,” *IEEE Network*, vol. 29, no. 5, pp. 36–42, 2015.

[12] W. Wei, H. Gu, K. Wang, X. Yu, and X. Liu, “Improving cloud-based iot services through virtual network embedding in elastic optical inter-dc networks,” *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 986–996, 2019.

[13] R. Zhu, S. Li, P. Wang, M. Xu, and S. Yu, “Energy-efficient deep reinforced traffic grooming in elastic optical networks for cloud–fog computing,” *IEEE Internet of Things Journal*, vol. 8, no. 15, pp. 12410–12421, 2021.

[14] S. Miladić-Tešić, G. Marković, D. Peraković, and I. Cvitić, “A review of optical networking technologies supporting 5g communication infrastructure,” *Wireless Networks*, vol. 28, no. 1, pp. 459–467, 2022.

[15] A. A. Barakabitze and R. Walshe, “Sdn and nfv for qoe-driven multimedia services delivery: the road towards 6g and beyond networks,” *Computer Networks*, vol. 214, Article ID 109133, 2022.

[16] J. D. Downie, X. Liang, and S. Makovejs, “Modeling the techno-economics of multicore optical fibers in subsea transmission systems,” *Journal of Lightwave Technology*, vol. 40, no. 6, pp. 1569–1578, 2022.

[17] C. Papapavlou, K. Paximadis, D. Uzunidis, and I. Tomkos, “Toward sdm-based submarine optical networks: a review of their evolution and upcoming trends,” *Tele.com*, vol. 3, no. 2, pp. 234–280, 2022.

[18] K. Saitoh and S. Matsuo, “Multicore fiber technology,” *Journal of Lightwave Technology*, vol. 34, no. 1, pp. 55–66, 2016.

[19] I. Brasileiro, L. Costa, and A. Drummond, “A survey on crosstalk and routing, modulation selection, core and spectrum allocation in elastic optical networks,” 2019, https://arxiv.org/abs/1907.08538.

[20] Z. Luo, S. Yin, L. Jiang, L. Zhao, and S. Huang, “Routing, spectrum and core assignment based on auxiliary matrix in the intra data center networks using multi-core fibers with super channel,” in *Proceedings of the 2020 Asia Communications and Photonics Conference (ACP) and International Conference on Information Photonics and Optical Communications (IPOC)*, pp. 1–3, Beijing, China, October 2020.

[21] R. Llorente, V. Fito, and M. Morant, “Optical combs and multicore fiber as technology enablers for next-generation datacenter infrastructure,” in *Metro and Data Center Optical Networks and Short-Reach Links V*, A. K. Srivastava, M. Glick, and Y. Akasaka, Eds., vol. 12027, Bellingham, WA, USA, International Society for Optics and Photonics, Article ID 120270E, 2022.

[22] H. Tode and Y. Hirota, “Routing, spectrum and core assignment for space division multiplexing elastic optical networks,” in *Proceedings of the 2014 16th International Telecommunications Network Strategy and Planning Symposium (Networks)*, pp. 1–7, Funchal, Portugal, September 2014.

[23] S. Fujii, Y. Hirota, T. Watanabe, and H. Tode, “Dynamic spectrum and core allocation with spectrum region reducing costs of building modules in aod nodes,” in *Proceedings of the 2014 16th International Telecommunications Network Strategy and Planning Symposium (Networks)*, pp. 1–6, Funchal, Portugal, September 2014.

[24] H. M. N. S. Oliveira and N. L. S. da Fonseca, “Protection, routing, spectrum and core allocation in eons-sdm for efficient spectrum utilization,” in *Proceedings of the 2019 IEEE International Conference on Communications (ICC)*, pp. 1–6, Shanghai, China, May 2019.

[25] A. Samuel, Y. Zhang, and R. Zhu, “Deadline-aware multicast resource allocation in sdm-eons with fluctuating delay-sensitive traffic,” *Journal of Lightwave Technology*, vol. 40, no. 16, pp. 5355–5368, 2022.

[26] J. Žerovnik, “Heuristics for np-hard optimization problems: simpler is better,” *Logistics & Sustainable Transport*, vol. 6, no. 1, pp. 1–10, 2015.

[27] X. Chen, B. Li, R. Proietti, H. Lu, Z. Zhu, and S. J. B. Yoo, “Deeprmsa: a deep reinforcement learning framework for routing, modulation and spectrum assignment in elastic optical networks,” *Journal of Lightwave Technology*, vol. 37, no. 16, pp. 4155–4163, 2019.

[28] B. Tang, Y.-C. Huang, Y. Xue, and W. Zhou, “Heuristic reward design for deep reinforcement learning-based routing, modulation and spectrum assignment of elastic optical networks,” *IEEE Communications Letters*, vol. 26, no. 11, pp. 2675–2679, 2022.

[29] T. Panayiotou, M. Michalopoulou, and G. Ellinas, “Survey on machine learning for traffic-driven service provisioning in optical networks,” 2022, https://arxiv.org/abs/2209.05080.

[30] X. Chen, R. Proietti, C.-Y. Liu, Z. Zhu, and S. J. B. Yoo, “Exploiting multi-task learning to achieve effective transfer deep reinforcement learning in elastic optical networks,” in *Optical Fiber Communication Conference (OFC) 2020*Optical Society of America, Washington, DC, USA, 2020.

[31] C. Natalino and P. Monti, “The optical rl-gym: an open-source toolkit for applying reinforcement learning in optical networks,” in *Proceedings of the 2020 22nd International Conference on Transparent Optical Networks (ICTON)*, pp. 1–5, Bari, Italy, July 2020.

[32] B. Li and Z. Zhu, “Deepcoop: leveraging cooperative drl agents to achieve scalable network automation for multi-domainsd-eons,” in *2020 Optical Fiber Communications Conference and Exhibition*, pp. 1–3, OFC), 2020.

[33] P. Morales, P. Franco, A. Lozada et al., “Multi-band environments for optical reinforcement learning gym for resource allocation in elastic optical networks,” in *Proceedings of the 2021 International Conference on Optical Network Design and Modeling (ONDM)*, pp. 1–6, Gothenburg, Sweden, June 2021.

[34] N. E. D. E. Sheikh, E. Paz, J. Pinto, and A. Beghelli, “Multi-band provisioning in dynamic elastic optical networks: a comparative study of a heuristic and a deep reinforcement learning approach,” in *Proceedings of the 2021 International Conference on Optical Network Design and Modeling (ONDM)*, pp. 1–3, Gothenburg, Sweden, June 2021.

[35] X. Luo, C. Shi, L. Wang, X. Chen, Y. Li, and T. Yang, “Leveraging double-agent-based deep reinforcement learning to global optimization of elastic optical networks with enhanced survivability,” *Optics Express*, vol. 27, no. 6, pp. 7896–7911, 2019.

[36] R. Zhu, S. Li, P. Wang, L. Li, A. Samuel, and Y. Zhao, “Deep reinforced energy efficient traffic grooming in fog-cloud elastic optical networks,” in *Proceedings of the 2020 Optical Fiber Communications Conference and Exhibition (OFC)*, pp. 1–3, San Jose, CA, USA, March 2020.

[37] X. Tian, B. Li, R. Gu, and Z. Zhu, “Reconfiguring multicast sessions in elastic optical networks adaptively with graph-aware deep reinforcement learning,” *Journal of Optical Communications and Networking*, vol. 13, no. 11, pp. 253–265, 2021.

[38] R. Li, R. Gu, W. Jin, and Y. Ji, “Learning-based cognitive hitless spectrum defragmentation for dynamic provisioning in elastic optical networks,” *IEEE Communications Letters*, vol. 25, no. 5, pp. 1600–1604, 2021.

[39] R. Zhu, G. Li, P. Wang, M. Xu, and S. Yu, “Drl-based deadline-driven advance reservation allocation in eons for cloud–edge computing,” *IEEE Internet of Things Journal*, vol. 9, no. 21, pp. 21444–21457, 2022.

[40] C. Wang, N. Yoshikane, F. Balasis, and T. Tsuritani, “Deepcms3: a deep reinforcement learning framework for core, mode and spectrum sequential scheduling over optical transport network,” in *Proceedings of the 2020 European Conference on Optical Communications (ECOC)*, pp. 1–4, Brussels, Belgium, December 2020.

[41] S. S. Mousavi, M. Schukat, and E. Howley, “Deep reinforcement learning: an overview,” in *Proceedings of the SAI Intelligent Systems Conference (IntelliSys) 2016*, Y. Bi, S. Kapoor, and R. Bhatia, Eds., Springer International Publishing, London, UK, pp. 426–440, September 2018.

[42] Y. Xiong, Y. Yang, Y. Ye, and G. N. Rouskas, “A machine learning approach to mitigating fragmentation and crosstalk in space division multiplexing elastic optical networks,” *Optical Fiber Technology*, vol. 50, pp. 99–107, 2019.

[43] Y. Xiong, Y. Ye, H. Zhang, J. He, B. Wang, and K. Yang, “Deep learning and hierarchical graph-assisted crosstalk-aware fragmentation avoidance strategy in space division multiplexing elastic optical networks,” *Optics Express*, vol. 28, no. 3, pp. 2758–2777, 2020.

[44] Q. Yao, H. Yang, R. Zhu et al., “Core, mode, and spectrum assignment based on machine learning in space division multiplexing elastic optical networks,” *IEEE Access*, vol. 6, pp. 15898–15907, 2018.

[45] V. Mnih, K. Kavukcuoglu, D. Silver et al., “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[46] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, *An Introduction to Deep Reinforcement Learning*, vol. 1, Now Foundations and Trends, Hanover; MA, USA, 2018.

[47] B. Kozicki, H. Takara, Y. Sone, A. Watanabe, and M. Jinno, “Distance-adaptive spectrum allocation in elastic optical path network (slice) with bit per symbol adjustment,” in *Proceedings of the 2010 Conference on Optical Fiber Communication (OFC/NFOEC)*, pp. 1–3, San Diego, CA, USA, March 2010.

[48] A. Muhammad, G. Zervas, and R. Forchheimer, “Resource allocation for space-division multiplexing: optical white box versus optical black box networking,” *Journal of Lightwave Technology*, vol. 33, no. 23, pp. 4928–4941, 2015.

[49] Y. Zhao, Y. Zhu, C. Wang et al., “Super-channel oriented routing, spectrum and core assignment under crosstalk limit in spatial division multiplexing elastic optical networks,” *Optical Fiber Technology*, vol. 36, pp. 249–254, 2017.

[50] I. Brasileiro, L. Costa, and A. Drummond, “A survey on challenges of spatial division multiplexing enabled elastic optical networks,” *Optical Switching and Networking*, vol. 38, Article ID 100584, 2020.

[51] C. Chen, M. Ju, S. Xiao, F. Zhou, and X. Yang, “Minimizing total blocking by setting optimal guard band in nonlinear elastic optical networks,” in *Proceedings of the 2017 19th International Conference on Transparent Optical Networks (ICTON)*, pp. 1–4, Bari, Italy, July 2017.

[52] A. Hill, A. Raffin, M. Ernestus et al., “Stable baselines,” 2018, https://github.com/hill-a/stable-baselines.

[53] V. Mnih, A. P. Badia, M. Mirza et al., “Asynchronous methods for deep reinforcement learning,” 2016, https://arxiv.org/abs/1602.01783.

[54] Y. Wu, E. Mansimov, S. Liao, R. Grosse, and J. Ba, “Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation,” 2017, https://arxiv.org/abs/1708.05144.

[55] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” 2017, https://arxiv.org/abs/1707.06347.

[56] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, “Trust region policy optimization,” in *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, pp. 1889–1897, Lille, France, December 2015.

[57] D. L. Mills and H. Braun, “The nsfnet backbone network,” in *Proceedings of the ACM Workshop on Frontiers in Computer Communications Technology, SIGCOMM ’87*, pp. 191–196, Association for Computing Machinery, Stowe, Vermont, August 1987.

[58] P. Batchelor, B. Daino, P. Heinzmann et al., “Study on the implementation of optical transparent transport networks in the european environment—results of the research project

cost 239," *Photonic Network Communication*, vol. 2, pp. 15–32, 2000.

[59] M. Klinkowski and G. Zalewski, "Dynamic crosstalk-aware lightpath provisioning in spectrally-spatially flexible optical networks," *Journal of Optical Communications and Networking*, vol. 11, no. 5, pp. 213–225, 2019.

[60] G. M. Saridis, D. Alexandropoulos, G. Zervas, and D. Simeonidou, "Survey and evaluation of space division multiplexing: from technologies to optical networks," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2136–2156, 2015.

[61] S. Fujii, Y. Hirota, and H. Tode, "Dynamic resource allocation with virtual grid for space division multiplexed elastic optical network," in *Proceedings of the 39th European Conference and Exhibition on Optical Communication (ECOC 2013)*, pp. 1–3, London, UK, September 2013.

[62] A. A. Saleh and J. M. Simmons, "Technology and architecture to enable the explosive growth of the internet," *IEEE Communications Magazine*, vol. 49, no. 1, pp. 126–132, 2011.

[63] J. Suárez-Varela, P. Almasan, M. Ferriol-Galmés et al., "Graph neural networks for communication networks: context, use cases and opportunities," 2021, https://arxiv.org/abs/2112.14792.