Nonlinear Impairment-Aware RMSA Under the Sliding Scheduled Traffic Model for EONs Based on Deep Reinforcement Learning

Yucong Zou, Xiaofeng Cai, Min Zhu, *Member, IEEE*, Jiahua Gu, *Graduate Student Member, IEEE*, Yunwu Wang, *Graduate Student Member, IEEE*, Guo Zhao, Chenglin Shi, Jiao Zhang, *Member, IEEE*, Yuancheng Cai, *Member, IEEE*, and Mingzheng Lei

***Abstract*—The elastic optical network (EON) can accommodate dynamic and diverse demands of next-generation applications by provisioning flexible lightpaths for them. Consequently, blocking probability of these requests can be reduced especially in sliding scheduled traffic, where requests are scheduled to start well after their arrival. In this article, a nonlinear impairment-aware routing, modulation and spectrum assignment (NLI-RMSA) problem was investigated and modeled under a sliding scheduled traffic model. We proposed a novel dynamic resource allocation scheme in EONs based on deep reinforcement learning (DRL) to jointly determine the path, modulation format and scheduling time of the lightpath requests. Several designed methods consider the nonlinear impairment in the physical layer to accurately estimate the channel state and increase the feasibility of the allocation scheme. Simulation results on the 14-node NSFNET topology demonstrate that the proposed DRL-based method greatly outperforms the two baseline heuristics, potentially saving at least 38.7% more blocking probability than the baseline heuristics.**

***Index Terms*—Blocking probability, deep reinforcement learning, nonlinear impairment, slide scheduling.**

I. Introduction

With the rapid growth of Internet traffic volume including ultra-high definition video, cloud computing and data center, there is a significant challenge faced by network infrastructures to support quality-of-transmission (QoT)-guaranteed, low-latency, and non-disruptive service for users [1]. Traditional wavelength-division multiplexing (WDM) optical networks follow the fixed uniform spacing and grid typically at 50 or 100 GHz, which will lead to the waste of optical spectrum. Recently, elastic optical networks (EONs) have emerged as a promising technology to accommodate dynamic and diverse demands of next-generation applications by adaptively allocating resources [2]. EON has a spectrum slot (e.g., 6.25 or 12.5 GHz), which is much finer than that of a WDM optical network and can also flexibly combine the slots to create super channels to serve the lightpath requests (LRs) [3]. To provision an LR in EON, three rules need to be followed, which are spectrum continuality, contiguity, and non-overlapping constraints [4]. Due to these constraints, the resource allocation in EONs becomes more complex. Hence, how to design a highly efficient LR provision scheme in EON becomes challengeable.

Many researches have been dedicated to resource allocation in EONs based on predefined transmission-reach (TR) limits, i.e., each modulation format has a corresponding TR limit, which requires a large margin for the most connections to guarantee QoT in the worst case through resource overprovisioning. In the result, it causes inefficient resource utilization [5]. Hence, some studies have been proposed to significantly improve spectral efficiency by accurately accounting for physical-layer impairments, using the Gaussian Noise (GN) model in EONs, instead of the existing TR-based methods [5], [6]. Recently, a transmission model is proposed by taking into consideration the nonlinear impairment (NLI) effect [5]. The NLI includes the additive spontaneous emission (ASE), self-channel interference (SCI), and cross-channel interference (XCI). ASE describes the noise due to the fiber amplifiers. SCI is the NLI produced by the channel onto itself. XCI is the NLI produced by the non-linear interaction of two channels [7]. The impairment-aware dynamic resource allocation in EONs includes two challenges: 1) the routing and spectrum allocation (RSA), where the spectrum continuity and contiguity constraints [8] are considered; and 2) the modulation format assignment accounting for the NLI. This is called the RMSA problem designed for nonlinear EONs.

In EONs, optical fibers transfer various kinds of traffic requests over different periods of time. In the static traffic model, the routing, modulation and spectrum allocation (RMSA)

Manuscript received 9 May 2023; revised 3 July 2023; accepted 24 July 2023. Date of publication 27 July 2023; date of current version 16 November 2023. This work was supported in part by the National Natural Science Foundation of China under Grants 62271135, 62101121, and 62101126, in part by the Key Research and Development Program of Jiangsu Province under Grant BE2020012, in part by the Major Key Project of Peng Cheng Laboratory under Grant PCL 2021A01-2, and in part by Project funded by China Postdoctoral Science Foundation under Grants 2021M702501 and 2022T150486. *(Corresponding author: Min Zhu.)*

Yucong Zou, Chenglin Shi, Yuancheng Cai, and Mingzheng Lei are with the Purple Mountain Laboratories, Nanjing 211111, China (e-mail: zouyucong@pmlabs.com.cn; cls@seu.edu.cn; caiyuancheng@pmlabs.com.cn; leimingzheng@pmlabs.com.cn).

Xiaofeng Cai, Min Zhu, Jiahua Gu, Yunwu Wang, and Jiao Zhang are with the National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China, and also with the Purple Mountain Laboratories, Nanjing 211111, China (e-mail: caixiaofeng@seu.edu.cn; minzhu@seu.edu.cn; gujiahua@seu.edu.cn; wangyunw@seu.edu.cn; jiaozhang@seu.edu.cn).

Guo Zhao is with the Nanjing Wasin Fujikura Optical Communication LTD, Nanjing 211111, China (e-mail: guo_zhao@nwf.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JLT.2023.3299272.

Digital Object Identifier 10.1109/JLT.2023.3299272

problems can be formulated since all the demands are known in advance. However, in the dynamic traffic model, the requests arrive and expire on-the-fly, so the problems become more challenging. In some dynamic circumstances, a service provider should provide bandwidth to satisfy the communication requirements of a company in a specific time. Hence, a traffic model is required to capture the characteristics of requests that require capacity during specific time intervals [9]. In contrast to on-demand immediate reservation services, the requests enjoy the efficiency of resource allocation because the holding duration can slide in a time window [10]. Since the requests only last during a specific time, they are dynamic in nature, which requires scheduled dedicated channels arranged by algorithms.

Recently, deep reinforcement learning (DRL) has demonstrated beyond human-level performance for resource allocation problems. DRL can learn successful policies progressively without any prior knowledge of the target system's behavior, by accumulating action experiences from repeated interactions with the target systems and reinforcing actions leading to higher rewards [11]. Moreover, DRL can support a variety of optimization objectives just by setting different reinforcement rewards. Therefore, we utilize DRL to conduct NLI-RMSA.

Our main contributions can be summarized as follows. 1) This article, to the best of our knowledge, is the first attempt to address an NLI-aware dynamic RMSA issue in EONs based on DRL method, which jointly allocates modulation formats and routes to minimize the blocking probability. 2) The effect of the physical-layer impairment is considered by utilizing the NLI model. Moreover, we use sliding scheduled traffic model to capture the characteristics of requests that require capacity during specific time intervals. Specifically, our designed algorithm can determine which LR should be served at first, and which route and modulation format should be assigned to each LR. 3) We design a DRL-based algorithm to achieve lower LR block probability, by introducing a time-frozen scheme to keep the action space small. Also, a heuristic method using priority queue is adopted to effectively assist the DRL agent in making decision. 4) Extensive numerical simulations are conducted to evaluate the superiority of our proposed DRL-based NLI-RMSA. Note that the two heuristics (i.e., KSP-FF and KSP-BF) generally use the fixed policy to schedule the LRs, and cannot be adapted to different request states, while our proposed DRL-based NLI-RMSA can accommodate the variability of network states. Simulation results show that our DRL-based NLI-RMSA is capable of maintaining low blocking probability, low delay tolerance and high spectrum efficiency under sliding scheduled traffic model with different arriving rate and sliding factors.

The rest of the article is organized as follows. Section II presents the NLI-RMSA problem formulation. We present the DRL-based NLI-RMSA design in Section III. In Section IV, the performance evaluation is shown in detail. Finally, we conclude the article in Section V.

II. Related Work

In [6], the authors proposed a hybrid NLI estimation technique along with a sophisticated K-least congested path routing strategy to solve the NLI-aware resource allocation problem in dynamic EONs. To shrink margin and improve spectrum resource utilization in EON, a more accurate model is required to describe the relationship between modulation format and all factors affecting the signal quality. As mentioned before, the signal quality, i.e., signal to noise ratio (SNR) decides whether a modulation format can be adopted. Thus, with the NLI model, we are able to accurately calculate the SNR for each LR, and decide the modulation format. This is formulated as an NLI-RMSA problem, which is even more complicated than the aforementioned RMSA problem and requires an effective algorithm design.

Most existing related work on the sliding scheduled traffic model are based on WDM optical networks. Lightpath switching is taken into account [12] to improve resource consumption efficiency, assuming Advance Reservation (AR) traffic. In [13] using AR scheduling, the λ-switching concept was applied to limit the overhead of reconfiguring resources for scheduling efficiency. As introduced in the literatures [14], [15], The AR model can be generally classified into four types. 1) STSD demand: it specifies a start time and duration is denoted, 2) STUD demand: it specifies a start time but no duration, 3) UTSD demand: it specifies a duration but no start time, and 4) UTUD demand: it neither specifies a start time nor a duration. Note that the sliding scheduled traffic model considered in this article is just one type of the AR models, i.e., STSD, because it has a specified start time and duration. Virtual Machine (VM)-placement and routing problem have been investigated with power saving and acceptance ratio optimized based on sliding scheduled traffic model [16]. In WDM network scenario, traffic grooming [17] was studied with a sliding scheduled traffic model to greatly decrease energy consumption and blocking probability.

The authors of [11] present a DRL-based online RMSA framework for dynamic traffic demands in EONs, which just adopts the TR limits to determine the modulation format and does not consider the actual network state. Zhu et al. [18] proposed a deep reinforced deadline-driven allocation (DRDA) algorithm for AR services. It assigns resources in the time domain and frequency domain to reduce the average initial delay and blocking probability. The optimization was done without considering the modulation problem and nonlinear impairment of channels. Chen et al. [19] introduced MTL, which is a multi-task learning aided knowledge transferring scheme to train the DRL agent. It enables the agent to learn and transfer knowledge across RMSA and anycast service provisioning tasks. The authors in [20] studied virtual network functions provisioning in inter-data-center EONs, which achieves low IT and spectrum resource consumption. Xu et al. [21] utilized graph convolutional neural networks and the recurrent neural network to extract the feature of EONs to help DRL solve the routing and spectrum assignment problems. In [22], authors designed a DRL-based observer to select the duration of each service cycle in the service frame work. The tradeoff among blocking probability, resource utilization and the number of network reconfigurations was balanced. In [23], the authors proposed a DRL-based RMSA agent for EONs, and then extend it to multi-domain EONs.

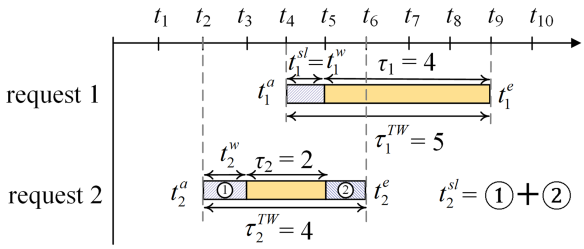


Fig. 1. Components of time window of LRs.

TABLE I
AVAILABLE MODULATION FORMATS AND THEIR LINEAR SCALE THRESHOLDS TO ACHIEVE A PRE-FEC BER OF 4 × 10^{-3} [24]

Modulation Format	m(bit/s/Hz)	SNR_{th}(dB)
PM-BPSK	2	3.52
PM-QPSK	4	7.03
PM-8QAM	6	17.59
PM-16QAM	8	32.60

III. PROBLEM FORMULATION

A. Substrate Network and Request Model

We model the EON as a directed graph G (V, E, F), where V and E are sets of nodes and fiber links, F denotes the frequency slot (FS) usage on each link $e \in E$. A sliding scheduled traffic model proposed in [9] is viable for the application in this article. A dynamic LR from source node o to destination node d with data rate b (Gbit/s), service duration τ, arriving time t_a and ending time t^e can be modeled as R (o, d, b, τ, t^a, t^e). In this model, the service duration τ is an interval within a time window [t^a, t^e], where the duration is allowed to slide. The actual starting time of an LR is variable relative to the left boundary t_a of the time window [t^a, t^e]. If an LR waits for t^w time units after t^a to start, the LR is active during the interval [t^a+t^w, t^a+t^w + τ]. As is shown in Fig. 1, the slack time $t^{sl} = t^e$ - t^a - τ is the maximum waiting time of an LR, which allows to postpone the service of the LR in order to accommodate more requests. The *slide factor* is defined as the slack time relative to the time window, i.e., s*f* = t^{sl}/τ^{TW}, where t^{sl} is the slack time and $\tau^{TW} = t^e$ - t^a is the time window duration. When $t^{sl} = 0$, it means that the starting time and ending time of the service duration are the same as the left and right boundary of the time window. In this case, the LR arrives and expires on-the-fly and needs to be served upon their arrivals. In the sliding scheduled traffic model, t^{sl} is usually larger than 0, it means that the starting and ending times of the service duration can be flexibly determined based on the current EON state. Benefiting from the flexibility, the service duration may slide earlier or later within the time window to avoid resource competition and reduce request blocking.

To serve request i, we need to compute an end-to-end path $p_{o,d}$, determine a QoT-guaranteed modulation format m, and allocate a set of spectrally continuous and contiguous FS's according to b and m on each fiber link along $p_{o,d}$. The number n_i of FS's allocated to request i can be formulated as [11],

$$n_i = \left\lceil \frac{b_i}{m_i \cdot B_{PM-BPSK}} \right\rceil \tag{1}$$

where BPM-BPSK is the data rate an FS can provide with modulation format binary phase shift keying (BPSK), and $m_i \in M = \{2, 4, 6, 8\}$ denotes the spectral efficiencies of Polarization multiplexing binary phase shift keying (PM-BPSK), PM-quadrature phase shift keying (QPSK), PM-8 quadrature amplitude modulation (QAM) and PM-16QAM respectively. Since the spectral efficiencies $m \in M$ are different, we can also use m_i to denote the corresponding modulation format assigned to connection of request i. For each available m, its required minimum SNR threshold *SNRth m* for $m \in M$ under a certain pre-forward error correction (FEC) bit-error rate (BER) (4 × 10^{-3} in this article) is given in Table I [24].

B. Nonlinear Impairment

The connection QoT can be estimated based on the GN model [24], which is an analytical model to calculate NLIs in dispersion-uncompensated links. By combing various NLIs including amplified spontaneous emission (ASE) noise, self-channel interference (SCI) and cross-channel interference (XCI), we can calculate the SNR for each connection of request i as,

$$SNR_i = \frac{G_i^{PSD}}{G_i^{ASE} + G_i^{SCI} + G_i^{XCI}} \tag{2}$$

where $G_i^{PSD}, G_i^{ASE}, G_i^{SCI}$, and G_i^{XCI}denotes power spectral density, amplified spontaneous emission noise, self-channel interference, and cross-channel interference of connection of request i respectively. They can be calculated as,

$$G_i^{ASE} = \left(e^{\alpha L} - 1\right) hvn^{sp} N_i^{span} \tag{3}$$

$$G_i^{SCI} = \frac{3\gamma^2 G^3}{2\pi\alpha \left|\beta_2\right|} \ln\left(\frac{\pi^2 \left|\beta_2\right|}{\alpha} \Delta f_i^2\right) N_i^{span} \tag{4}$$

$$G_i^{XCI} = \frac{3\gamma^2 G^3}{2\pi\alpha \left|\beta_2\right|} \sum_{j \neq i} \ln\left|\frac{\Delta f_{ij} + \Delta f_j/2}{\Delta f_{ij} - \Delta f_j/2}\right| N_{ij}^{span} \tag{5}$$

where α, β_2, γ, h, n^{sp}, v, L, Δf_i, Δf_{ij} denotes the power attenuation, the fiber dispersion, the fiber nonlinear coefficient, Planck's constant, the spontaneous emission factor, the optical carrier frequency and the length of each span, the bandwidth used by request i, the interval between the bandwidth occupied by request i and j, respectively. The number of spans propagated by connection of request i along the route is denoted by N_i^{span}. And N_{ij}^{span} denotes the number of spans shared by the routes of connection of request i and request j. Hence, these three equations suggest the interference encountered by each transmission channel, i.e., each allocated frequency slot, comes from the number of spans of the routes and the LRs in service. Specifically, the different spectrum location for the request i will cause a different XCI nonlinear impairment G_i^{XIC}, as indicated by (5). This, in turn, would affect the SNR of the request i (i.e., QoT).

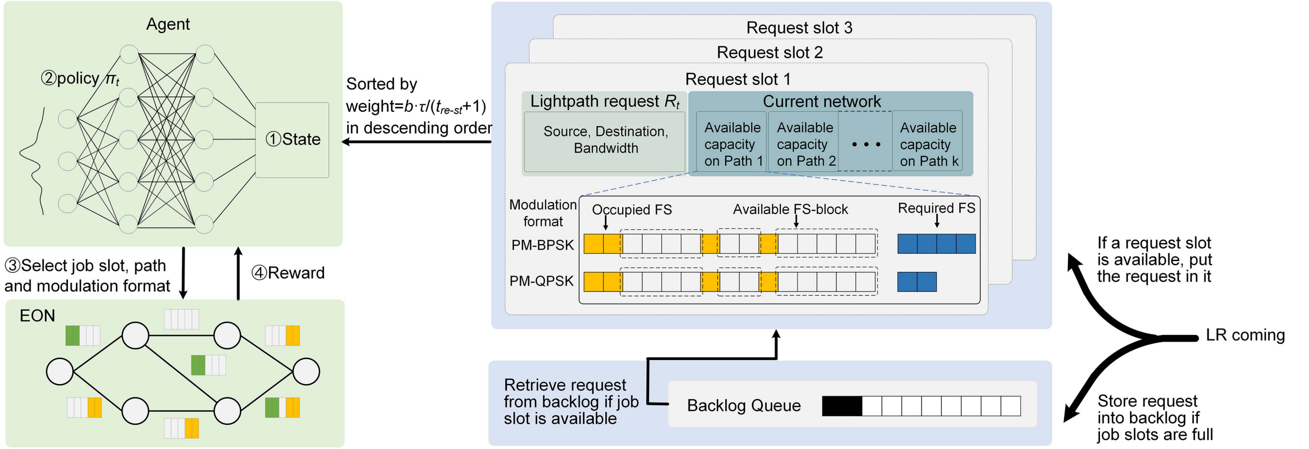

Agent
②policy π_t
①State
Sorted by
weight=$b\cdot\tau/(t_{re\text{-}st}+1)$
in descending order
③Select job slot, path and modulation format
④Reward
EON
Request slot 3
Request slot 2
Request slot 1
Lightpath request R_t
Source, Destination, Bandwidth
Current network
Available capacity on Path 1
Available capacity on Path 2
Available capacity on Path k
Modulation format
Occupied FS
Available FS-block
Required FS
PM-BPSK
PM-QPSK
If a request slot is available, put the request in it
LR coming
Store request into backlog if job slots are full
Retrieve request from backlog if job slot is available
Backlog Queue

Fig. 2. Operation principle of DRL-based NLI-RMSA.

Note that, in dynamic RMSA, requests arrive and expire on-the-fly and need to be served in the slack time upon their arrivals. Hence request i incurs XCI from not only the existing but also future LRs. But the XCI effect from the future LRs on request i is not available at the moment of the provisioning request i. Therefore, we assume that: 1) the number of the future LRs is estimated by multiplying the average arrival rate and the service duration of current R_i, 2) the bandwidth of the future LR is identical to the average bandwidth of all the LRs, 3) the number of FS's that is used by a future LR is calculated by applying the lowest modulation format, and 4) the allocated FS's for future LRs are placed next to the highest FS of current request i in the fiber links. Request i is served successfully only if its SNR satisfies the SNR threshold of the used m_i, i.e., $SNR_i \geq SNR_{th}$. New arriving LRs can be served upon their arrivals. If there are not enough resources to serve the LRs, they will await in RS request slots. The number of LRs beyond the first RS is counted in the backlog as shown in Fig. 2. We do not merge them into a single queue for the following three reasons. 1) If the request slots and backlog are merged into a single queue, the waiting queue becomes much longer to buffer the waiting requests. However, according to the comparison of blocking probability vs. number of request slots as shown in Fig. 6, we can find that the request blocking probability may not always decrease as the request slots increase. So, it is not urgent to provide too many request slots for the waiting requests. 2) The larger request slots may enlarge the action space of the DRL agent, which will inevitably increase the iteration times and deteriorate the learning efficiency. 3) A request that has been waiting for a longer time should be given a higher priority to be served by our proposed scheduling algorithm.

The notations of all variables and parameters are listed in Table II.

TABLE II
NOTATIONS FOR THE PARAMETERS AND VARIABLES

Parameters	Description
V	Set of nodes
E	Set of fiber links
F	Frequency slot (FS) usage on each link
o	Source node
d	Destination node
b	Bandwidth
τ	Service duration
t_a	Arriving time
t_e	Ending time
R	Set of LRs
t_s	Starting time
t_{sk}	Slack time
sf	Slide factor
$p_{o,\,d}$	an end-to-end path from o to d
m	Modulation format
R_i	the i-th request in R
n_i	Number of FS's allocated to R_i
b_i	Bandwidth of R_i
m_i	Modulation format of R_i
SNR_m^{th}	SNR threshold for m
G_i	Power spectral density of R_i
G_i^{ASE}	Amplified spontaneous emission noise of R_i
G_i^{SCI}	Self-channel interference of R_i
G_i^{XCI}	Cross-channel interference of R_i
α	Power attenuation
β_2	Fiber dispersion
γ	Fiber nonlinear coefficient
h	Planck's constant
n_{sp}	Spontaneous emission factor
v	Optical carrier frequency
L	the length of each span
N_{span}^i	Number of spans propagated by R_i
N_{span}^{ij}	Number of spans shared by the routes of R_i and R_j
SNR_i	SNR of R_i
M	Number of job slots

IV. DRL-Based NLI-RMSA Design

In this section, we first present the proposed DRL-based NLI-RMSA scheme in detail, which includes state representation, action space, reward, and DNNs. Following this, we elaborate the training process through the pseudocode of the training algorithm.

Fig. 2 illustrates the operation principle of the proposed DRL-based NLI-RMSA. The intelligent agent is presented as a DNN, which is also referred as a policy network. Upon request i arrives at time t, the policy network takes the state information s_t including the state of request i and current network as the input (*step* 1), and outputs the probability distribution π_t ($A|$ s_t, θ) over all possible action space A. And θ represents the policy parameters of the DNN (*step* 2). Based on the π_t, the DRL-based NLI-RMSA agent takes an action $a_t \in A$ and attempts to establish the ligthpath for request i (*step* 3). A reward r_t related to the NLI-RMSA operation is fed back to the agent (*step* 4). The r_t together with s_t and a_t, can be used to train the NLI-RMSA agent.

1) State Representation: The state representation s_t is an $1 \cdot [2 \cdot |V| + 1 + (2 \cdot J + 3) \cdot |M| \cdot K + 2]$ array. As is shown in Fig. 2, each LR in the job slot has a weight that can be calculated by $b \cdot\ \tau/(t_{re\text{-}st} + 1)$, where $t_{re\text{-}st}$ is the remaining slack time. Through a weighted ranking method, we will sequentially serve the requests in the request slot queue using the designed weight in descending order. Note that the weight consider the request's b, τ, and slack time ($t_{re\text{-}st}$) simultaneously. This is similar to the best-fit decreasing strategy in bin packing problems [25], where prioritizing larger requests can enhance resource utilization and lower request blocking rate. Meanwhile, we give priority to the requests with shorter slack times. This is because once the slack time is exceeded, these requests will be rejected, leading to a high blocking rate.

Fig. 2 shows the state representation s for the DRL-based NLI-RMSA, which includes the information of the selected LR and spectrum utilization on the K-shortest candidate paths. They are with J available FS-blocks based on $|M|$ different modulation formats. The array is defined as,

$$s_t = \{o, d, \tau, \{\{\{z^{1,j}_{k,m}, z^{2,j}_{k,m}\}|_{j\in[1,J]}, z^{3,j}_{k,m}, z^{4,j}_{k,m}, z^{5}_{k,m}\}|_{m\in[1,M]}\}|_{k\in[1,K]}, c^s, c^b\} \quad (6)$$

we use $2 \cdot |V| + 1$ elements to represent a LR from o to d in the one-hot format (or one-hot encoding) and service duration τ, where $|V|$ denotes the number of nodes in EON. For each k of the K candidate paths, we use M different modulation format to calculate $z^{3,j}_{k,m}$, which is the number of required FS's of a request based on modulation format m. $z^{1,j}_{k,m}, z^{2,j}_{k,m}, z^{4,j}_{k,m}$ and $z^{5}_{k,m}$ denotes the size of j-th available FS-block, the starting FS index of j-th available FS-block, the average size of J available FS-blocks and the total size of available FS-blocks on fiber links of the k-th path with m modulation format, respectively.

As Fig. 2 shows, if the agent selects the PM-QPSK instead of the PM-BPSK, the number of required FS's decrease (i.e., 4→2) and the number of the available FS-blocks increase accordingly (i.e., 2→3). Although a higher level of modulation format can save spectrum resources, its SNR requirement gets stricter (Table I). Therefore, we need to verify whether the connection that consists of the k-th path, modulation format m and j-th FS-block hypothetically-assigned to R_i can meet the corresponding SNR constraint. When the following three cases occur: 1) the number of available candidate paths between o and d is smaller than K, the number of available FS-blocks is smaller than J, 3) the pre-computed connection fails to satisfy the SNR constraint, we would assign an array of -1 for corresponding part to keep the format of state representation consistent. c^s describes the number of LRs in the request slot and c^b describes the number of LRs in the backlog.

*2) Action Space:*In our simulations, the whole time axis is divided in terms of slotted time. In each time slot, the DRL agent will schedule multiple lightpath requests in the request slot queue. Assume there are 'X' lightpath requests to be scheduled, the action space for the DRL will be a large action space of size $(K{\times}M{+}1)^X$, where K is the number of candidate paths, and M is the number of candidate modulation formats. This would make the DRL learning quite challenging. To avoid this, we reduce the action space by allowing the agent to execute multiple actions within a single time slot, where each action step is referred to as 'time-frozen step' [26]. In a frozen step, when an LR is selected, agent will assign a routing path from the K candidates, one of M modulation format and one of the J FS-blocks on the selected path. The action space is given by $\{\varnothing, (1, 1), (1, 2), \ldots, (1, |M|), (2, 1), \ldots, (K, |M|)\}$, where $a = (k, m)$ means schedule the LR in the k-th path with the modulation format m. And $a = \varnothing$ is a void action which means the agent will not schedule the LR. Once the LR is scheduled, the agent will remove the LR from the request slots. If the backlog is not empty, an LR is taken from its head and accommodated into the job slot. If the action is invalid (e.g., the selected path and modulation format does not meet the SNR requirement) or void, the agent will exit the current frozen step. The process is summarized by a flowchart in Fig. 3.

Note that the proposed DRL-based method is just used for the routing and modulation format allocation, while the spectrum slot resources are allocated by using the classical first-fit scheme. It is due to the following two reasons. 1) The first-fit scheme is often used in LR resource provision [1], [11], [19], which can reduce spectrum fragments and improve spectrum utilization. 2) To reduce the action space of the DRL agent and improve the learning efficiency, the DRL just takes charge of the decision of routing and modulation format, while ensuring the optimal allocation strategy.

3) Reward: After taking an action, the agent will receive a reward r_t immediately. The objective of the algorithm is to minimize the number of blocked requests. Thus, if the epoch is finished, $r_t =$ the total time and spectrum resource of successfully served LRs, i.e., $\sum_{i\epsilon R-R_{block}} b_i \cdot \tau_i$. Otherwise, $r_t = 0$. The reward is calculated at the end of the epoch and passed to each action in the process through policy gradient (PG) training, which guarantees global optimality.

The pseudocode of the training algorithm is presented in Algorithm 1. In line 1, the policy network is initialized. To ensure sufficient training, we randomly generated multiple sets of request sets and ran EP episodes for each set to obtain multiple sets of different environment-agent interaction samples (lines 2-5). At each time step, the agent will sequentially allocate the LR requests in the Request slots. Since the agent's allocation decisions for these requests are made within the same time step,

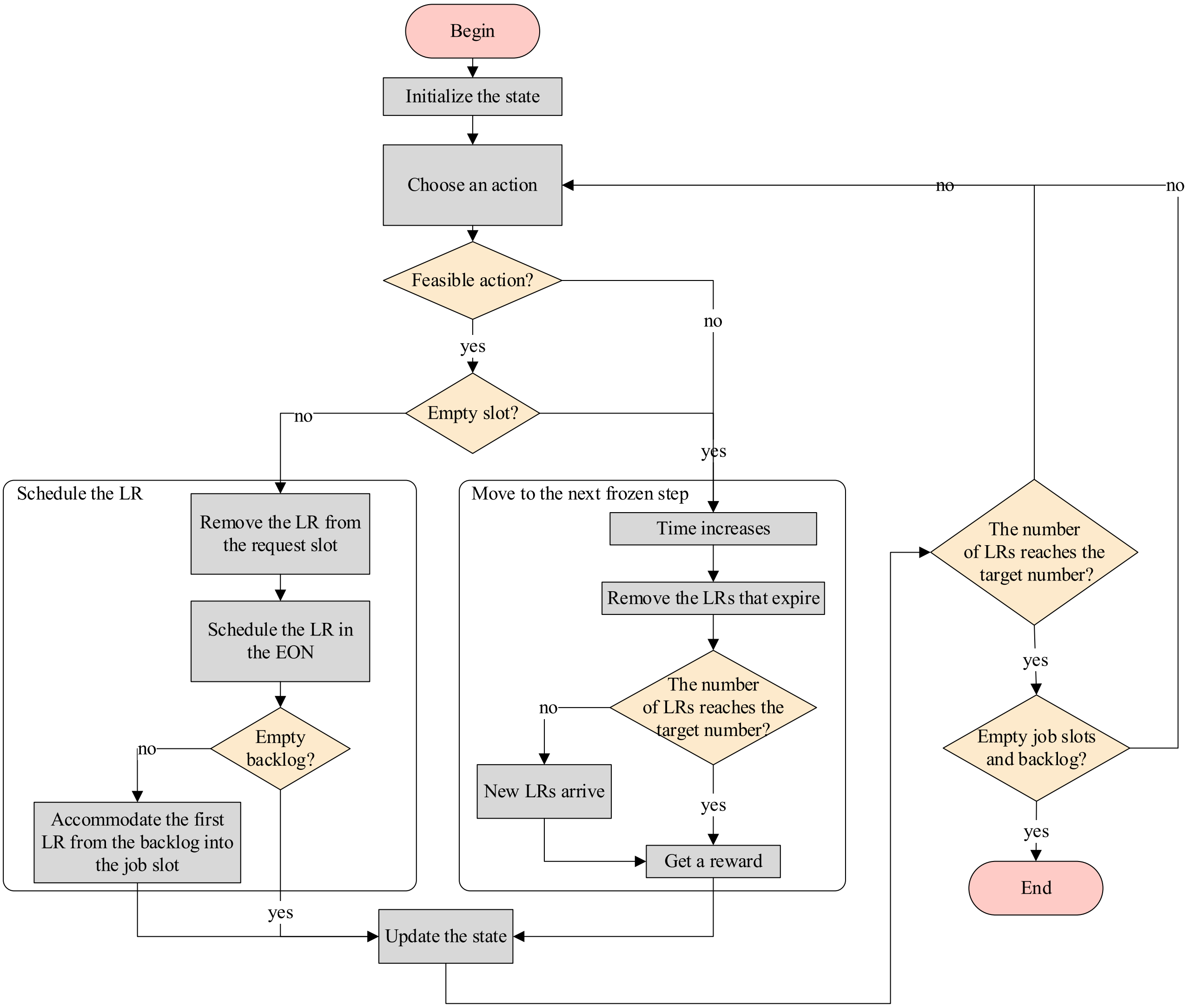

Begin
Initialize the state
Choose an action
Feasible action?
no
yes
Empty slot?
no
yes
Schedule the LR
Remove the LR from the request slot
Schedule the LR in the EON
Empty backlog?
no
Accommodate the first LR from the backlog into the job slot
yes
Move to the next frozen step
Time increases
Remove the LRs that expire
The number of LRs reaches the target number?
no
New LRs arrive
yes
Get a reward
Update the state
The number of LRs reaches the target number?
no
yes
Empty job slots and backlog?
no
yes
End

Fig. 3. The flowchart of the DRL-based NLI-RMSA.

we refer to this decision-making process as occurring within a ‘frozen time step’ [26]. If an invalid action is selected, the agent will exit the current frozen time step and proceed to the next time step. All environment-agent interactions (s,a,r) are stored in a buffer Λ_{ep} for subsequent updates to the neural network parameters θ (Lines 6-13). Once the termination condition is met (i.e., all requests have been allocated), the data collection for this training episode is completed (lines 14-17). Subsequently, we use the classical PG algorithm - REINFORCEMENT with baseline [26] - to calculate the return value v, baseline b, and the policy network parameter update $\Delta\theta$, and finally update θ (Lines 18-27).

V. Evaluation

A. Simulation Setup

The performance of the proposed DRL-based NLI-RMSA is evaluated in 14-node NSFNET topology in Fig. 4 (which is also used for performance evaluation in [11]). The parameters related to the physical impairments are $\alpha = 0.22$ dB/km, $\gamma = 1.3$ (W·km)$^{-1}$, $\beta_2 = -21.7$ ps^2/km, $n^{sp} = 1.58$, $v = 193.55$ THz, $L = 100$ km, respectively [24]. A uniform PSD $G^{PSD} = 15$ mw/THz is assumed for all connections. Each fiber link in substrate EON accommodates 100 FS’s of 12.5 GHz each. The dynamic LRs are generated with an average service duration τ_{ave} as 20 time units according to exponential distribution. The required data rate of each LR is distributed evenly with [80,

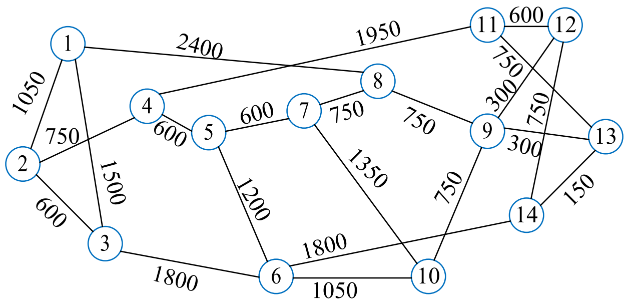


Fig. 4. 14-node NSFNET topology.

Algorithm 1: Pseudo-Code for the DRL-Based NLI-RMSA Training Algorithm.

```
Initialize policy network with $\theta$;
for each iteration do
  $\Delta\theta = 0$;
  for each request set do
    run episode $ep = 1, \ldots, EP$:
    set buffer $\Lambda_{ep} = \{\}$, and index in buffer $l_{ep} = 1$;
    for each time step $t$ do
    while true do
      collect the agent-environment interaction ($s$, $a$, $r$) in the current frozen step following the "Schedule the LR" branch in the Fig. 3. Assign ($s$, $a$, $r$) with index $l_{ep}$ (i.e.,$s^{ep}_{l_{ep}}$,$a^{ep}_{l_{ep}}$,$r^{ep}_{l_{ep}}$) and add them into $\Lambda_{ep}$. Then, $l_{ep} = l_{ep} + 1$.
      if an infeasible action is made or an empty slot is selected then
        break (proceed to the "Move to the next frozen step" branch in the Fig. 3);
      end
    end
    if the number of LRs reaches target number and request slot & back log are both empty then
      break;
    end
   end
  compute returns: $v^{ep}_l = \sum_{k=l}^{l_{ep}} \gamma^{k-l} r_k$;
  for $l = 1$ to $\max\{l_1, l_2, \ldots, l_{EP}\}$ do
  compute baseline: $b_l = \frac{1}{EP} \sum_{ep=1}^{EP} v^{ep}_l$;
    for $ep = 1$ to $E$ do
        $\Delta\theta = \Delta\theta + \alpha \nabla_\theta \log \pi_\theta(s^{ep}_l, a^{ep}_l)(v^{ep}_l - b_l)$;
      end
    end
   end
  $\theta = \theta + \Delta\theta$
end
```

320] Gb/s. The values of the slide factor are between 0.1 and 0.9. The simulation results are averaged after 10 times of operation.

If the mean duration of a request is τ_{mean} time units, the service rate μ corresponding to each request is $1/\tau_{mean}$. If the maximum number of simultaneously processed requests in the EON system is assumed to be '$N^{max\text{-}req}$', then the constraint for the system to maintain a stable state is $\lambda/(N^{max\text{-}req} \times \mu) < 1$ [27]. The traffic load for the EON system can be represented as $\rho = \lambda/\mu$ in Erlang, where λ is the mean arrival rate and μ is the service rate. Hence, the traffic load can essentially go up to $N^{max\text{-}req}$ Erlang (i.e., $\rho_{\max} = N^{max\text{-}req}$). According to Fig. 5, when $\rho = 350$ Erlang, the blocking probability is at least 0.2 for all algorithms, which has reached a higher level. Therefore, we set $N^{max\text{-}req} = 350$ and thus $\rho \in [100, 350]$ Erlang, so that $\lambda/(N^{max\text{-}req}\mu) < 1$ is satisfied to maintain the EON system stable.

In order to evaluate in detail the system performance with our proposed DRL algoritm, we define five metrics.

1) Blocking Probability: Considering the specific bandwidth and time requirements of different requests, we define the request blocking probability as follows [28],

$$P_{block} = \frac{\sum_{i \in R_{block}} b_i \cdot \tau_i}{\sum_{i \in R} b_i \cdot \tau_i} \tag{7}$$

where R is the set of all the requests, R_{block} is the set of the blocked requests, b_i is the required data rate of request i, and τ_i is the service duration of request i.

2) Spectrum Efficiency: Since different requests would adopt different modulation formats with different spectrum utilization, the average spectral efficiency is defined as follows,

$$\eta = \frac{\sum_{i \in (R - R_{block})} (b_i \cdot \tau_i)}{\sum_{i \in (R - R_{block})} (n_i \cdot BW^{fs} \cdot \tau_i)} \tag{8}$$

where n_i is the number of FS's occupied by request i, and BW^{fs} is the bandwidth of an FS (i.e., 12.5 GHz).

To evaluate the system delay performance, we define the following metrics: including the delay tolerance, delay cost, and contribution of zero delay requests.

3) Delay Tolerance: For the accepted requests, the average delay tolerance (DT) is defined in (9).

$$T_{DT}^{R-R_{block}} = \frac{\sum_{i \in (R - R_{block})} \left(t_i^w / \tau_i^{TW}\right)}{|R - R_{block}|} \tag{9}$$

where t_i^w is the waiting time for request i to be deployed, τ_i^{TW} is the duration of the time window of request i.

4) Delay Cost: Given the different durations τ of different requests, it is expected that the requests with the smaller τ wait for less time to be deployed, while the requests with the larger τ would wait for the longer time. Therefore, the average delay cost (DC) of the accepted requests is defined as follow,

$$T_{DC} = \frac{\sum_{i \in (R - R_{block})} \left(t_i^w / \tau_i\right)}{|R - R_{block}|} \tag{10}$$

5) Contribution of Zero Delay Requests: In the sliding scheduled request model, we also calculate the contribution of zero delay requests in respect to all accepted requests, which is defined as follow.

$$P_{ZD} = \frac{|R_{zero}|}{|R - R_{block}|} \tag{11}$$

where R_{zero} is the set of requests that is deployed on arriving without any delay.

B. Blocking Probability

In Fig. 5, we observe that the blocking probability increases as the traffic load increases. It is primarily due to the increasing resource requirement and decreasing SNR of the LRs as the traffic load increases. High load means that more requests hold the resources for a longer time. As a result, other requests are unable to be allocated in time. The optical channel gets crowded with more requests, and some requests have to take a longer detour, which brings worse nonlinear impairment and leads to the further SNR degradation. In turn, it is inevitable to increase the request blocking probability. From Fig. 5, it is found that our proposed DRL-based NLI-RMSA outperforms the existing KSP-BF and KSP-FF algorithms. As previously described, our

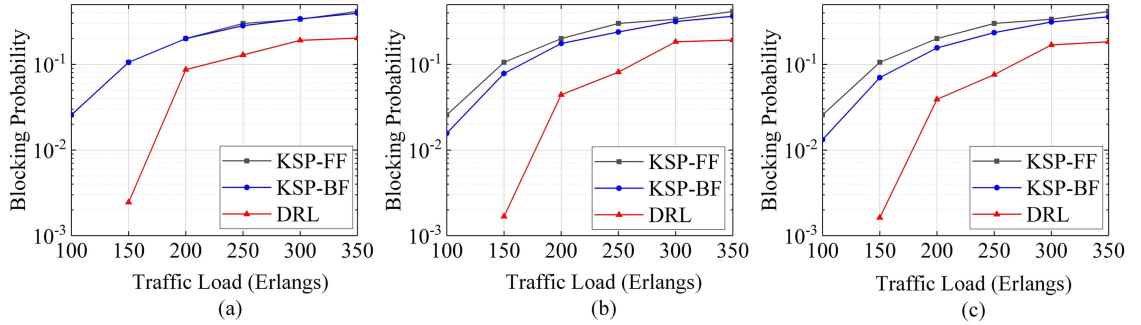


Fig. 5. Blocking probability when the number of request slots is (a) 1; (b) 5; (c) 9.

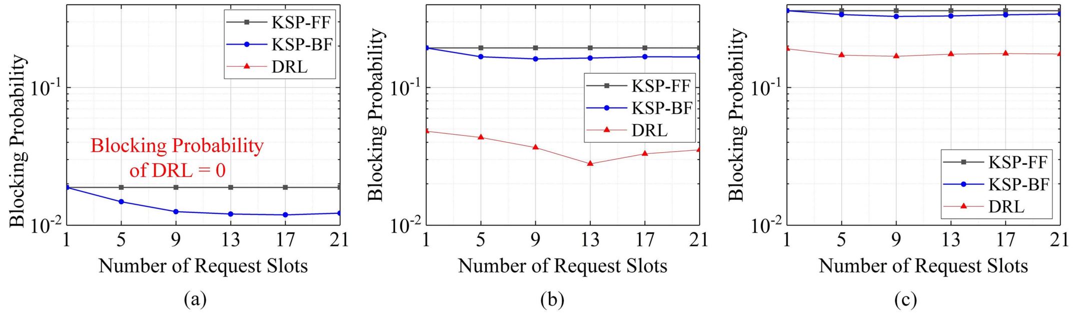


Fig. 6. Blocking probability *vs.* number of request slots under the different traffic loads of (a) 100 erlangs; (b) 200 erlangs; (c) 300 erlangs.

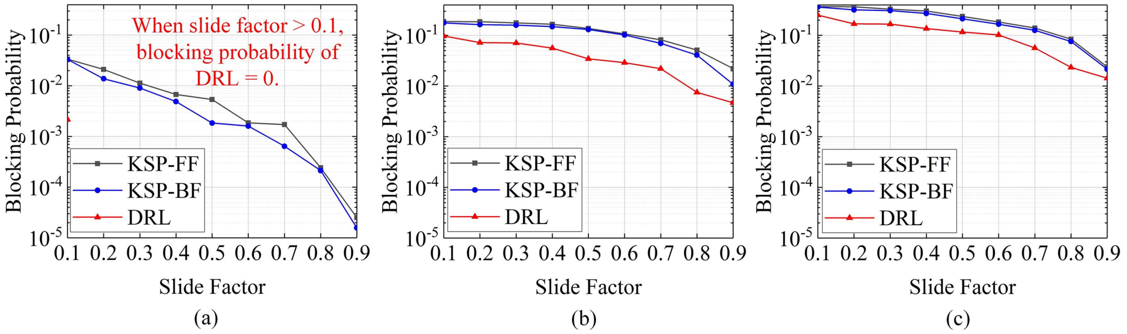


Fig. 7. Blocking probability *vs.* slide factor when the traffic load is (a) 100 erlangs; (b) 200 erlangs; (c) 300 erlangs.

proposed DRL method can flexibly accommodate the changes of the network states, while the heuristics just follow a fixed policy to schedule the dynamic LRs. It is noted that the KSP-BF is slightly better than the KSP-FF, just because that the KSP-BF sorts the LRs in the job slots. Thus, the LRs with larger bandwidth, longer duration time and shorter slack time will be given priority to be processed. In Fig. 5(a), when the number of job slots is 1, KSP-BF degenerates to KSP-FF, so the curves of KSP-FF and KSP-FF coincide. It is also noted that when the number of job slots are increased from 1 to 5, there is a decrease in the DRL blocking probability. But as the number of job slots are further increased from 5 to 9, the DRL blocking probability appears to remain unchanged.

In Fig. 6, when the number of request slots increases, the KSP-FF blocking probability remains unchanged, because it does not sort the requests, which are processed just in an arriving order. But for the DRL-based and KSP-BF algorithms, the LR blocking probabilities first decrease slightly and then remain unchanged as request slots increases. This is because that the LR sorting between some request slots might produce some optimization effects. However, the dynamic arriving LRs may not fill in all the available. So the blocking probability may not always decrease, as the request slots increases.

In Fig. 7, we evaluate the effect of the slide factor on the LR blocking probability. As the slide factor increases, it means that the incoming LR has a relatively longer slack time before the LR's expiration, and thus it can wait in the job slots and backlog until enough the route and spectrum resources are released for the LR deployment. The results in Fig. 7 also illustrate that our proposed DRL-based method can outperform the KSP-BF and KSP-FF algorithms, due to the obtained optimal provision via the DRL iterative leaning.

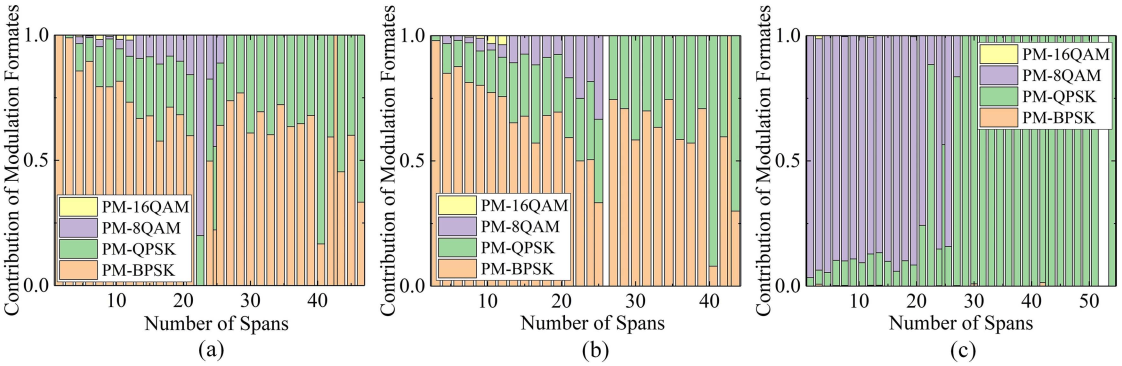


Fig. 8. Contribution of modulation formats of (a) KSP-FF; (b) KSP-BF; (c) DRL.

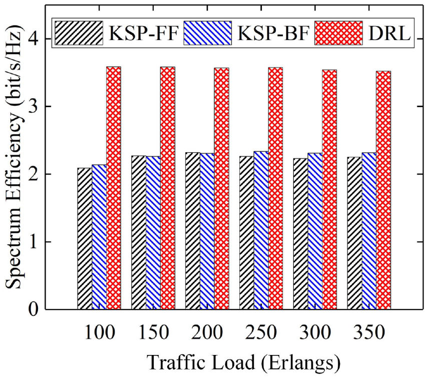


Fig. 9. Spectrum efficiency vs. traffic load.

C. Modulation Formats and Spectrum Efficiency

Fig. 8 statistics the contributions of the modulation formats at the different numbers of spans for all the accepted LRs with the three algorithms, respectively. In Fig. 8(a) and (b), both two heuristics (i.e., KSP-FF and KSP-BF) mainly uses PM-BPSK and PM-QPSK. The higher-order PM-8QAM and PM-16QAM account for a smaller proportion and are mainly distributed for the lightpath length shorter than 25 spans. It is because that the two heuristics would select the modulation format from low to high order. But the DRL often selects the most suitable format via the iterative learning. Thus, it not only saves the spectrum resource consumption, but also satisfies the SNR requirement of the transmission lightpath. In Fig. 8(b), we can see that the light-path length of 26 spans was not used by any LR. It is because that the light-path length in network topology are discrete, and hence it is not necessary that any light-path lengths are used. The same phenomenon occurs in Fig. 8(c), where the bar at 52 spans is missing due to the same reason.

In Fig. 9, we can observe the average spectrum efficiency of the DRL-based NLI-RMSA is better than two heuristics under different traffic loads. It is because the proposed DRL method can choose flexibly the most suitable modulation format according to the current network state. The flexibility allows DRL to prefer choosing those higher-order modulation formats with the higher spectrum efficiency, while avoiding the transmission nonlinear impairment, which is also consistent with Fig. 8.

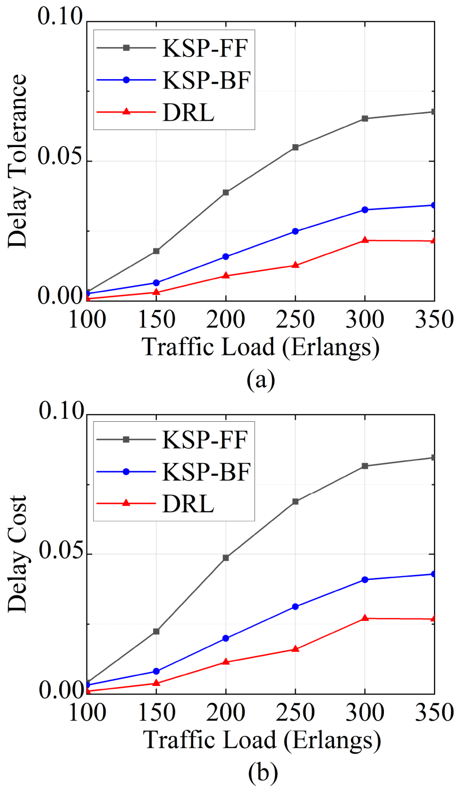


Fig. 10. Simulation results on delay: (a) Delay tolerance; (b) delay cost.

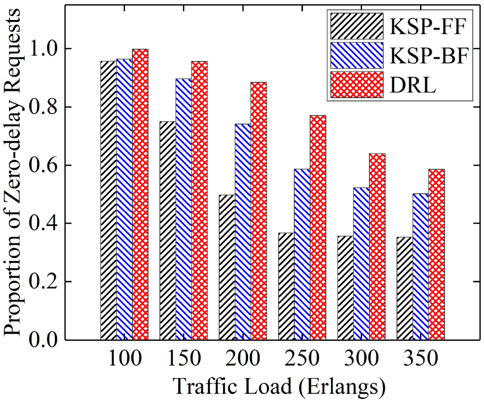
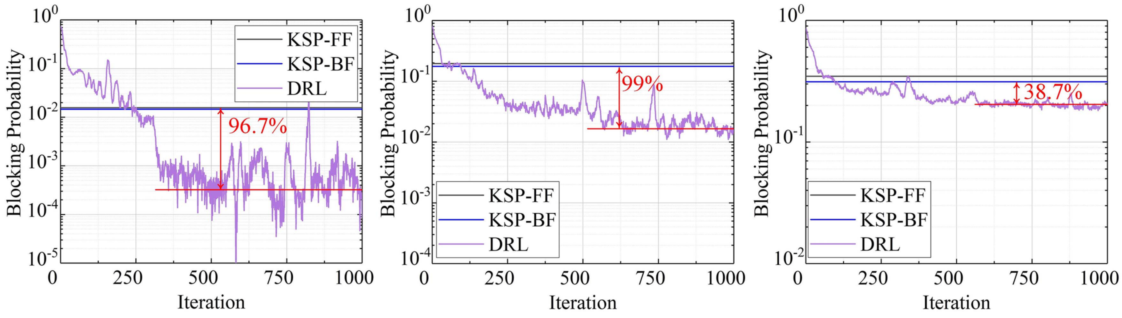


Fig. 11. Proportion of zero-delay requests.



(a) (b) (c)

Fig. 12. Training curve of DRL when the traffic load is (a) 100 erlangs; (b) 200 erlangs; (c) 300 erlangs.

D. Delay Tolerance and Delay Cost

Fig. 10 shows the simulation results related to the system delay performances. In Fig. 10(a), the delay tolerance of three algorithms increases as the traffic load increases. It is because that when the traffic load is smaller, the spectrum resource is sufficient in the EON, and hence the LRs can be deployed without waiting for a long time. However, as the traffic load gets larger, the spectrum resource is in high demand. In this case, the arriving requests have to wait until the occupied resource are released again. Thus, it is inevitable that the average ratio (i.e., delay tolerance) of the required waiting time to the total time window increases. From Fig. 10(a), the delay tolerance of the DRL-based algorithm is better than two other heuristics, because the DRL-based algorithm can efficiently allocate the required spectrum resources to accommodate more LRs, and thus the LR's waiting time becomes less. Similarly, in Fig. 10(b), the delay cost of three algorithms increases as the traffic load increases. The delay cost paid by the DRL-based algorithm is the smallest. The reason is similar to that for delay tolerance.

Fig. 11 shows the simulation results for the proportion of zero-delay requests. With the increase of traffic load, for three algorithms adopted, the proportion of the zero-delay decreases. This is because the increase of traffic load will lead to the decrease of the available resources in the EONs, and hence all the arriving LRs may not be deployed immediately. Compared with the two heuristics, the DRL-based algorithm achieves a higher proportion of the zero-delay requests. This is because that by using DRL method, the efficient utilization of spectrum resources can greatly reduce the congestion in the time domain and allows the arriving LRs to be deployed immediately.

E. DRL Training Evaluation

In Fig. 12(a), the blocking rate of the DRL-based algorithm is much lower than the two heuristics when it is trained after 250 epochs. Specifically, it converges to about 0.0005 when it is trained until 300 epochs. Compared with the KSP-FF and KSP-BF, the proposed DRL-based algorithm can achieve a considerable drop of 96.7%, 99% and 38.7%, respectively, when the traffic load is 100, 200 and 300 Erlangs. It is due to the fact that these two heuristics ignore the flexibility of choosing different paths and modulation formats, and just operates in a first-fit way. For the DRL-based algorithm, it can achieve optimal policy with the help of DRL iteration. Different from the KSP-FF that prefers the first arriving LRs, the DRL-based algorithm selects the LRs to be scheduled considering the remaining slack time. Moreover, the DRL-based algorithm can achieve a balance between bandwidth consumption and slack time to reduce the blocking probability. Note that, in principle, the DRL interacts with the environment to learn optimal action strategies for the maximum reward. Thus, the DRL indeed needs a period of time for training. However, given request traffic model and system transmission capacity, the proposed DRL method can be trained offline. After the iterative learning, we can obtain the optimal scheduling strategy for our NLI-RMSA scheduling problem, which is stored in advance in the network controller. In other words, the actual duration of the DRL training will not affect the real-time decision making and the system performance.

VI. Conclusion

In this article, we propose a DRL-based NLI-RMSA scheme to solve dynamic sliding scheduled LR provisioning problem in EONs. We introduce a nonlinear model to estimate the physical layer impairments of the transmission channel. A sliding scheduled traffic model is utilized to capture the characteristics of LRs that require capacity during specific time intervals. The proposed three algorithms can determine the start time, routing and modulation formats of the arriving LRs. The extensive simulation results show that by varying the traffic load and slide factor, the proposed DRL-based algorithm can significantly reduce the system blocking probability and delay of requests, and achieve better spectrum efficiency, compared with the two heuristic baselines.

References

[1] Y. Lei, Q. Chen, Y. Jiang, Q. Zhang, and B. Chen, "Crosstalk-aware routing, core, and spectrum assignment with core switching in SDM-EONs," in *Proc. Int. Conf. Opt. Commun. Netw.*, 2019, pp. 1–3.

[2] S. Liu, W. Lu, and Z. Zhu, "On the cross-layer orchestration to address IP router outages with cost-efficient multilayer restoration in IP-over-EONs," *J. Opt. Commun. Netw.*, vol. 10, no. 1, pp. A122–A132, Jan. 2018.

[3] J. Zhao, H. Wymeersch, and E. Agrell, "Nonlinear impairment-aware static resource allocation in elastic optical networks," *J. Lightw. Technol.*, vol. 33, no. 22, pp. 4554–4564, Nov. 2015.

[4] G. Savva, K. Manousakis, and G. Ellinas, "Eavesdropping-aware routing and spectrum/code allocation in OFDM-based EONs using spread spectrum techniques," *J. Opt. Commun. Netw.*, vol. 11, no. 7, pp. 409–421, Jul. 2019.

[5] L. Yan, E. Agrell, H. Wymeersch, and M. Brandt-Pearce, "Resource allocation for flexible-grid optical networks with nonlinear channel model," *J. Opt. Commun. Netw.*, vol. 7, no. 11, pp. B101–B108, 2015.

[6] R. Wang, S. Bidkar, F. Meng, R. Nejabati, and D. Simeonidou, "Load-aware nonlinearity estimation for elastic optical network resource optimization and management," *J. Opt. Commun. Netw.*, vol. 11, no. 5, pp. 164–178, 2019.

[7] P. Poggiolini, G. Bosco, A. Carena, V. Curri, Y. Jiang, and F. Forghieri, "The GN-model of fiber non-linear propagation and its applications," *J. Lightw. Technol.*, vol. 32, no. 4, pp. 694–721, Feb. 2014.

[8] B. C. Chatterjee, N. Sarma, and E. Oki, "Routing and spectrum allocation in elastic optical networks: A tutorial," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 3, pp. 1776–1800, Third Quarter 2015.

[9] B. Wang, T. Li, X. Luo, and Y. Fan, "Traffic grooming under a sliding scheduled traffic model in WDM optical networks," in *Proc. IEEE Workshop Traffic Grooming WDM Netw.*, 2004, pp. 1–10.

[10] S. Zhang, D. Shen, and C.-K. Chan, "Energy-efficient traffic grooming in WDM networks with scheduled time traffic," *J. Lightw. Technol.*, vol. 29, no. 17, pp. 2577–2584, Sep. 2011.

[11] 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," *J. Lightw. Technol.*, vol. 37, no. 16, pp. 4155–4163, Aug. 2019.

[12] P. Afsharlar, J. M. Plante, A. Deylamsalehi, and V. M. Vokkarane, "Delayed wavelength switching and allocation in optical networks," in *Proc. IEEE Sarnoff Symp.*, 2018, pp. 1–6.

[13] J. M. Plante and V. M. Vokkarane, "Sliding scheduled lightpath establishment with time-slotted wavelength-switching," *J. Opt. Commun. Netw.*, vol. 9, no. 1, pp. 119–137, 2017.

[14] J. Zheng and H. T. Mouftah, "Supporting advance reservations in wavelength-routed WDM networks," in *Proc. IEEE Int. Conf. Comput. Commun. Netw.*, 2001, pp. 594–597.

[15] N. Charbonneau and V. M. Vokkarane, "A survey of advance reservation routing and wavelength assignment in wavelength-routed WDM networks," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 4, pp. 1037–1064, Fourth Quarter 2012.

[16] A. Dalvandi, M. Gurusamy, and K. C. Chua, "Power-efficient and predictable data centers with sliding scheduled tenant requests," in *Proc. IEEE Int. Conf. Cloud Comput. Technol. Sci.*, 2014, pp. 547–554.

[17] X. Zhang and L. Xu, "Energy-efficient traffic grooming under sliding scheduled traffic model for IP over WDM optical networks," *China Commun.*, vol. 11, no. 7, pp. 74–83, Jul. 2014.

[18] 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 Things J.*, vol. 9, no. 21, pp. 21444–21457, Nov. 2022.

[19] X. Chen, R. Proietti, C.-Y. Liu, and S. J. B. Yoo, "A Multi-task-learning-based transfer deep reinforcement learning design for autonomic optical networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 9, pp. 2878–2889, Sep. 2021.

[20] M. Zhu, Q. Chen, J. Gu, and P. Gu, "Deep reinforcement learning for provisioning virtualized network function in inter-datacenter elastic optical networks," *IEEE Trans. Netw. Serv. Manage.*, vol. 19, no. 3, pp. 3341–3351, Sep. 2022.

[21] L. Xu, Y.-C. Huang, Y. Xue, and X. Hu, "Deep reinforcement learning-based routing and spectrum assignment of EONs by exploiting GCN and RNN for feature extraction," *J. Lightw. Technol.*, vol. 40, no. 15, pp. 4945–4955, Aug. 2022.

[22] B. Li, W. Lu, and Z. Zhu, "Deep-NFVOrch: Leveraging deep reinforcement learning to achieve adaptive vNF service chaining in DCI-EONs," *J. Opt. Commun. Netw.*, vol. 12, no. 1, pp. A18–A27, Jan. 2020.

[23] B. Li, R. Zhang, X. Tian, and Z. Zhu, "Multi-agent and cooperative deep reinforcement learning for scalable network automation in multi-domain SD-EONs," *IEEE Trans. Netw. Serv. Manage.*, vol. 18, no. 4, pp. 4801–4813, Dec. 2021.

[24] L. Yan, E. Agrell, M. N. Dharmaweera, and H. Wymeersch, "Joint assignment of power, routing, and spectrum in static flexible-grid networks," *J. Lightw. Technol.*, vol. 35, no. 10, pp. 1766–1774, May 2017.

[25] K. Wang, W. Zhou, and S. Mao, "On joint BBU/RRH resource allocation in heterogeneous cloud-RANs," *IEEE Internet Things J.*, vol. 4, no. 3, pp. 749–759, Jun. 2017.

[26] H. Mao, M. Alizadeh, I. Menache, and S. Kandula, "Resource management with deep reinforcement learning," in *Proc. 15th Assoc. Comput. Machinery Workshop Hot Topics Netw.*, 2016, pp. 50–56.

[27] S. Das and M. Ruffini, "A variable rate fronthaul scheme for cloud radio access networks," *J. Lightw. Technol.*, vol. 37, no. 13, pp. 3153–3165, Jul. 2019.

[28] W. Lu and Z. Zhu, "Dynamic service provisioning of advance reservation requests in elastic optical networks," *J. Lightw. Technol.*, vol. 31, no. 10, pp. 1621–1627, May 2013.