Actor-Critic-Based Resource Allocation for Multi-modal Optical Networks

1st Boyuan Yan, Yongli Zhao, Yajie Li, Xiaosong Yu, and Jie Zhang
Beijing University of Posts and Telecommunications
Beijing, China
{yanboyuan, yonglizhao, yajieli, xiaosongyu, lgr24}@bupt.edu.cn

Ying Wang, Longchun Yan
State Grid Information & Telecommunication Company
Beijing, China
1829523652@qq.com

Sabidur Rahman
University of California at Davis
Davis, CA, USA
krahman@ucdavis.edu

***Abstract*—With the rapid development of optical network, network status appears more and more features. For example, flexi-grid technology introduces noticeable features about spectral constraints and deep features about spectral fragmentation. However, limited by complex non-linear relationships among different features and optimization objectives, traditional heuristic algorithms for resource allocation cannot discover and utilize proper combination of these features sometimes. Reinforcement learning (RL) is an autonomic learning technology that could dig out essential features automatically for network optimization with different objectives. In this paper, we introduce the concept of multi-modal optical networks to represent different features of optical networks, and propose actor-critic-based resource allocation (ACRA) algorithm to improve the performance of resource allocation in optical networks. Simulation results show that multi-modal representation method can accelerate the learning efficiency, and the proposed ACRA algorithm can achieve the optimization of resource allocation.**

Keywords—optical networks, resource allocation, reinforcement learning, actor-critic, multi-modal learning

I. INTRODUCTION

Last two decades witness the explosive growth of global internet traffic. The transmission data of global Internet network increases about 23 million times from 1992 to 2016, and will grow at a compound annual growth rate of 24% from 2016 to 2021 [1]. Meanwhile, various emerging network services, such as augmented reality (AR), virtual reality (VR), and internet of things (IoT), will require more bandwidth with various requirements. Driven by the traffic growth and variety of services, the scale of optical networks continuously extends in transport networks and access networks, and optical transmission technology has evolved from wavelength division multiplexing (WDM) to space division multiplexing (SDM). This makes routing calculation and resource allocation much more complex. Many heuristic algorithms are proposed to allocate bandwidth resource for different optimization objectives by considering different critical factors in optical networks. First-Fit (FF) [2], random pick [3-4], and best-fit strategies [5-6] are proposed to allocate wavelength with the purpose of reducing blocking probability by considering the order of wavelengths. Q-factor [7], bit error rate (BER) [8], and optical signal to noise ratio (OSNR) [9] are considered in order to guarantee the quality of service. Shared risk link group (SRLG) [10] and protecting spanning tree set [11] are used to improve the survivability of optical networks.

However, there are two key challenges for these algorithms. Firstly, RA heuristic algorithm may need to take a huge amount of efforts to seek the proper relationships among those factors and the goals, such as the relationship between evaluation of spectral fragmentation and improvement of bandwidth utilization. Secondly, there may be many other potential factors which prior studies did not consider. Reinforcement Learning (RL) is able to solve these two challenges. The basic idea of RL is to capture the most important influence factors of the real problem facing a learning agent interesting with its environment to achieve a goal [12]. In general, The agent tries millions of times to seek a proper policy to solve the problem. With each one of a series of attempts, the agent perceives the environment, obtains the feedback for previous action, and take an action to change or keep the state of environment. The input data has significant effects on the learning process. Firstly, the input data determines the best performance that agent could reach, because all the information comes from it. Secondly, the input data is also able to affect the learning speed and direction, because different expressions for the same data have different feature distributions. Special design for data representation is a feasible way to control the influence of the input data . In this paper, we propose the concept of multi-modal optical networks to control the representation for optical networks. Then, we propose a RL algorithm to allocate resource efficiently in optical networks, named actor-critic-based resource allocation (ACRA) algorithm.

The rest of this paper is organized as follows. Section **II** introduces the background of RL. Section **III** explains the definition of multi-modal optical networks. Section **IV** proposes actor-critic-based RA (ACRA) algorithm. Section **V** shows simulation results of the proposed algorithm. Section **VI** draws a conclusion.

II. REINFORCEMENT LEARNING BACKGROUND

In general, RL is composed of agent, environment, and the closed loop of interactive process between them. At every time step $t+1$, agent gets the reward r_t for previous action a_t, and observes the current state s_{t+1} of environment ε. Then agent takes an action a_{t+1} from all possible actions A according to policy π, where π is a mapping from s_{t+1} to a_{t+1}. The environment changes to s_{t+1} by the action, and generates reward r_{t+1} for the agent. Then agent obtains new state and reward, and starts a new loop. The process continues until the terminal state is triggered. The goal of RL is to maximize the accumulated return from time t, which is $R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$ with discount factor $\gamma \in (0,1]$. RL consists of value-based model-free methods and policy-based model-free methods roughly. In value-based RL like Q-learning [13], maximum return is achieved by updating action-value

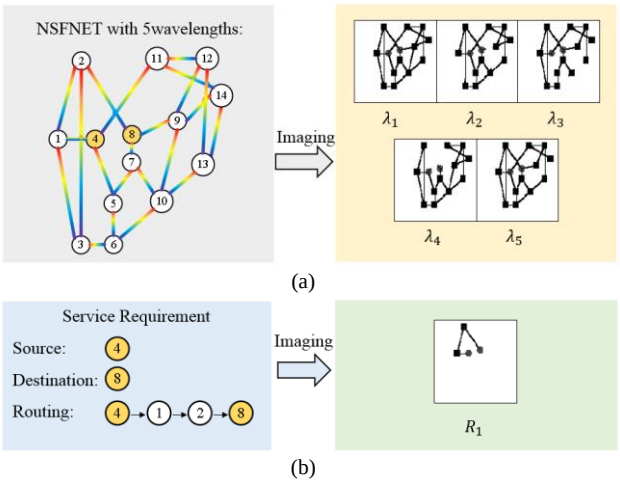


Fig. 1 Illustration of multi-modal optical network. (a) basic topology modality; (b) basic routing modality

function $Q(s,a;\theta)$ under state s and action a with parameter θ to approximate an optimal function $Q^*(s,a)=\mathrm{E}_{\max}[R_t \mid s_t=s,a]$. In policy-based RL like REINFORCE family [14], the policy is parameterized directly as $\pi(a \mid s;\theta)$ with parameter θ, whose updating is to improve $E[R_t]$ for maximum return.

The action-value function and policy functions have been implemented in the existing researches. However, deep convolutional neural network (CNN) becomes the mainstream since Alpha-Go obtains an absolute advantage in Go community [15-16]. CNN is a kind of feed-forward artificial neural network (ANN) by introducing convolutional layer. CNN is a key technology in artificial intelligence (AI), which plays a key role in reinforcement learning, computer vision, natural language recognition, and other fields of AI. In this paper, our proposed ACRA algorithm uses CNN to allocate bandwidth resource in wavelength division multiplex (WDM) networks.

III. Multi-modal Optical Networks

Information in real world usually comes through multiple modalities. Each modality follows distinct statistical properties, and different modalities typically carry different kinds of information [17]. For example, in optical networks, adjacency list shows the connections between different optical node pairs, but OSNR shows the transport attribute of each optical link. Besides, the same information could also be carried in different modalities in different ways. Routing table that contains all feasible routing paths includes routing information obviously. And adjacency list also contains routing information implicitly, because it could be used to calculate routing from one to another by using routing algorithm.

How to design the proper format to represent the changing state of optical networks as the input is a key issue for the application of RL algorithm. In this paper, multiple modalities are introduced into optical networks in order to help agent ignore the interference and focus on the information that is strongly correlated to the objective. We design two basic modalities for WDM networks, i.e., basic topology modality and basic routing modality (see Fig. 1). In Fig. 1(a), the NSFNET topology with five max-available wavelengths is converted to five sub-graphs. Each one of them indicates the topology on specific wavelength, which may be different from each other. Besides, incoming service request can also be converted to sub-graphs, named basic routing modality in Fig. 1(b). The requirements of service contain only the source node and destination node in optical networks, for simplicity. A routing path $4\rightarrow1\rightarrow2\rightarrow8$ is calculated by a routing algorithm. Then, there are four wavelength choices for this routing apart from the fourth wavelength topology. The selection depends on the output of RL algorithm.

Both modalities are implemented in the form of graphs. In the next sections, the graph of different modalities is 8-bit grayscale with 112x112 pixels. The reason of imaging is that topology data of optical networks is a non-Euclidean space, just like sensor networks [18]. However, deep neural network is proven to be a powerful tool for these fields with Euclidean or grid-like structure like computer vision, but not non-Euclidean structure. So, imaging is used to convert network from non-Euclidean space to Euclidean space. Besides, the content augmentation on image does not need change its form, and the process to handle this input would not be modified in some situation.

IV. Acror-Critic-Based Resource Allocation

A. Procedure of ACRA algorithm

In this paper, we propose an actor-critic-based resource allocation (ACRA) algorithm to solve routing and wavelength assignment problem. The cooperation of actor

TABLE I. Pseudo-Code of ACRA

ACRA Algorithm	
1.	The instance starts, all wavelengths of optical network in this instance are available.
2.	$t \leftarrow 1, l_v \leftarrow 0, l_a \leftarrow 0, e \leftarrow 0$
3.	**While** $t < T$ **do**:
4.	$t_s \leftarrow t$
5.	Observe state s_t to check the occupation of optical network and if a service request arrives.
6.	**While** $s_t \notin S$ and $t - t_s \neq U$ **do**:
7.	Take an action a_t to allocate wavelengths of available routing path according to policy function $\pi(a_t \mid s_t;\theta)$.
8.	Observe state s_{t+1} and get reward r_{t+1}.
9.	$t \leftarrow t+1$
10.	**If** $s_t \in S$ **do**:
11.	$R_t \leftarrow 0$
12.	Clean all occupation of wavelengths to reset optical network.
13.	**Else do**:
14.	$R_t \leftarrow V(s_t;\theta_v)$
15.	**For** $i \leftarrow t-1, t-2, \ldots, t_s$ **do**:
16.	Calculate return $R_i \leftarrow r_i + \gamma \cdot n_{i+1}$
17.	Calculate loss l_v, l_a, and entropy e.
18.	Calculate total loss $l_t = l_v \cdot c_l + l_a - e \cdot c_e$
19.	Update θ_v and θ_a according to loss l_t
20.	Instances ends.

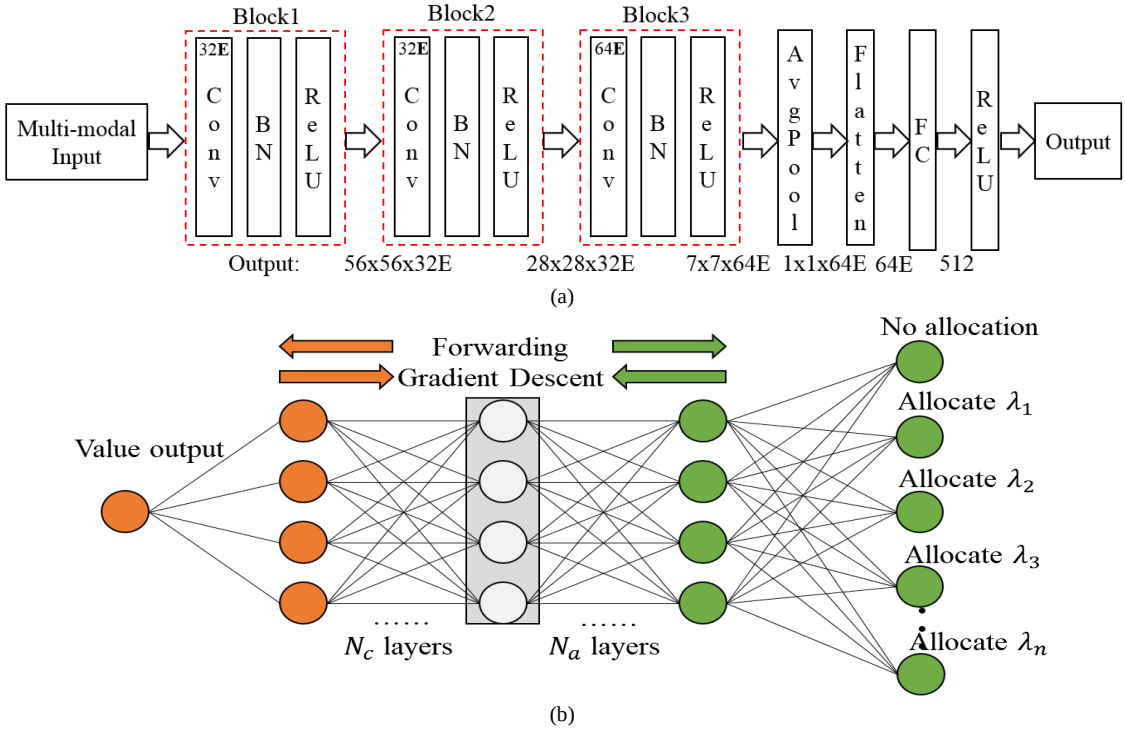


Fig. 2 Neural networks design of actor and critic: (a) Extensible simple network; (b) branches of critic and actor network

and critic [19] is introduced into RL for performance improvement. In ACRA, actor is responsible for selecting an available wavelength or do nothing for service requirement, and critic is responsible for evaluating the state of optical networks, and could assist actor to extract essential features. In order to formulate the process of ACRA, some variables and functions are defined. t is the time step in the whole process of simulation that starts from 1. U is update frequency for policy. T is the total number of steps. S is the state set that indicates the instance is terminated and will restart in next time step. ACRA uses policy function $\pi(a|s;\theta_a)$ with parameter θ_a as the actor, and value function $V(s;\theta_v)$ with parameter θ_v as the critic [20]. l_v is the loss for critic, and l_a is the loss for actor. e is the entropy of the probability distribution for each action calculated by $\pi(a|s;\theta_a)$. l_t is the total loss for critic, actor and the entropy.

In ACRA, multiple instances run simultaneously in a multi-thread way. Data from different instances would be collected while actor and critic need update, and parameters keep synchronization during simulation. Tab. I shows the procedure of ACRA algorithm in a single instance. When an instance starts, all parameters in ACRA are initialized, and all resources are not occupied in optical networks at Step 1. For every discrete time step from Step 6 to Step 9, current state of two basic modalities is observed, and the reward is received. If wavelength allocation is completed in optical networks or any service arrives, the state is changed. When the received state belongs to S, or time interval reaches U, the update is triggered at Step 10. The update for θ_a and θ_v uses the data of state, action and reward. Firstly, the return is calculated from time step $t-1$ to t_s in the reversed order from Step 10 to Step 16. Then l_v and l_a are calculated at Step 17, according to Eq. (1) and (2). l_v is the mean square error (MSE) between the return and value function. l_a is the cross entropy between policy function and the difference of return and value function. Then, the total loss is calculated at Step 18 by using l_v, l_a, and e, as Eq. (3) shows. c_l and c_e are the coefficients to adjust the weight of l_v, l_a, and e. Finally, we update θ_a and θ_v with the gradient descent method at Step 19, which is the most important method in neural network. The introduction of entropy e aims to evaluate and increase the possibility difference of actions.

$$l_v = \frac{1}{t-t_s}\sum_{i=t_s+1}^{t}(R_i - V(s_i;\theta_v))^2 \quad (1)$$

$$l_a = -\frac{1}{t-t_s}\sum_{i=t_s+1}^{t}(R_i - V(s_i;\theta_v))\cdot\log(\pi(a_i|s_i;\theta_a)) \quad (2)$$

$$l_t = l_v\cdot c_l + l_a - e\cdot c_e \quad (3)$$

B. *Design of actor and critic*

In ACRA, both actor and critic are designed as ANN. A shared common CNN structure named extensible shared network (ESNet) is designed to extract the key features from multi-modal input. As Fig. 2 (a) shows, ESNet is a simple CNN structure like AlexNet [21]. ESNet contains three similar stacked blocks that contains convolutional layer (Conv), batch normalization layer (BN), and rectified linear unit layer (ReLU). The first two Conv have the same parameters where kernel size equals 3. The stride is 2, and padding is 1. The last Conv has the parameter where kernel size equals 5. The stride is 4, and padding is 1. The kernel numbers of three Conv are 32E, 32E, and 64E respectively, where **E** is a zoom factor for the width of ESNet. According to convolution implementation by Pytorch, the relationship between the input size and output size of Conv is as Eq. (4).

So the output of each block is $56\times56\times32E$, $28\times28\times32E$, and $7\times7\times64E$. Then an average pooling layer (AvgPool) is behind these blocks where kernel size equals 7. The padding is 0, and stride is 1. Then the output of AvgPool compresses the scale of previous output to $1\times1\times64E$, and a flatten layer reduce the useless dimensions to just $64E$. Finally, a fully connected layer (FC) and a ReLU is designed to summary features from $64E$ kernels jointly.

$$output = \left\lfloor \frac{input + 2 \cdot padding - kernel\_size}{stride} + 1 \right\rfloor \quad (4)$$

Fig. 2(b) shows the following fully-connected neural network structure with two branches, where each circle stands for a neuron. The layer with a grey background in the middle is the input layer of this network, also the output of ESNet. The left branch with N_c+1 FC layers is the rest of $V(s;\theta_v)$ that outputs value, and the right branch with N_a+1 layers is the rest of $\pi(s;\theta_a)$ that outputs probability distribution for every possible action. The direction from the middle to the two sides shows the forwarding calculation, and the reversed direction from the sides to the middle shows the gradient descent for parameter θ_v and θ_a.

C. Reward and Punishment Mechanism

ACRA contains a reward and punishment mechanism (RPM) to control the feedback of environment, which is the mapping from impact of action to signed number. Both short term and long term effects need to be taken into consideration. For different objectives, the designs of RPMs should also be different. For example, the action that reroutes original path to a shorter path should be rewarded for improving resource utilization, but maybe punished for lower BER.

Dynamic resource allocation of optical networks could be considered as a discrete time Markov process. At any time step, in order to improve resource utilization, the decision simply depends on that if there is a service, and if there exists available wavelength for the arrived service. Tab. II shows the reward and punishment for different actions under different conditions. There are only two conditions that should be rewarded. If there is no service arrival at some time step, agent should take no action to avoid disrupt the environment. And if there exist available wavelengths for arrived service, agent should allocate resource for the service requirement.

TABLE II. REWARD AND PUNISHMENT

Condition		Action	Reward/Punishment
Service Arrival	**Wavelength Available**		
No	--	No Action	R_1
		Others	P_1
Yes	No	No Action	P_2
		Others	P_3
	Yes	No Action	P_4
		Unavailable wavelength	P_5
		Available wavelength	R_2

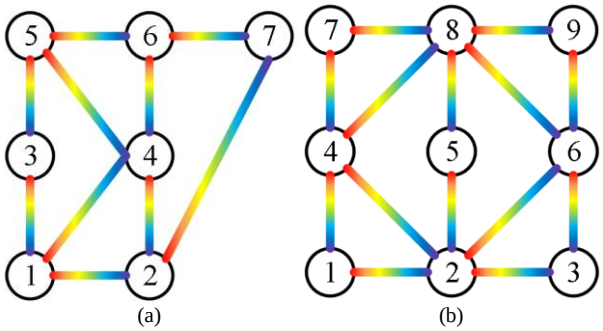
V. SIMULATION RESULT

In this section, we evaluate the performance of ACRA, and evaluate the effectiveness of multi-modal input by simulation. We use two topologies of Fig. 3 and NSFNET of Fig. 1 as the simulated WDM networks. The number of parallel instances implemented are 128. The update interval U is 5, so the batch size of input is 640. The total number of steps T for every instance is 1×10^7 that takes about 5 days on our machine without GPU. Discount factor γ is 0.99, that means the rewards of adjacent steps have similar influence. c_l is 0.5 and c_e is 0.01 [22]. In RPM, R_1 is 0, R_2 is 1, and all feedbacks of punishment are -1. In the process of gradient descent, we use RMSProp optimizer [23] where basic learning rate equals $7e-4$, factor ε equals $1e-5$, and factor α equals 0.99. There is no learning rate adjustment in simulation, because learning rate is small enough. Both N_c and N_a are 0 for simplicity. In order to avoid unnecessary influence while imaging WDM networks, we firstly adjust the relative locations of optical nodes to make links easy to distinguish on graphs. And keep these locations in two modalities in simulation.

Limited by the hardware, the number of available wavelength is set as 5. Both the arrival and the leave of service follow discrete Poisson processes. The source node and destination node of service are generated randomly. Dijkstra shortest path (DSP) algorithm is used to calculate routing, which is the data of basic routing modality. DSP and FF algorithm (DSP-FF) is used as the benchmark.

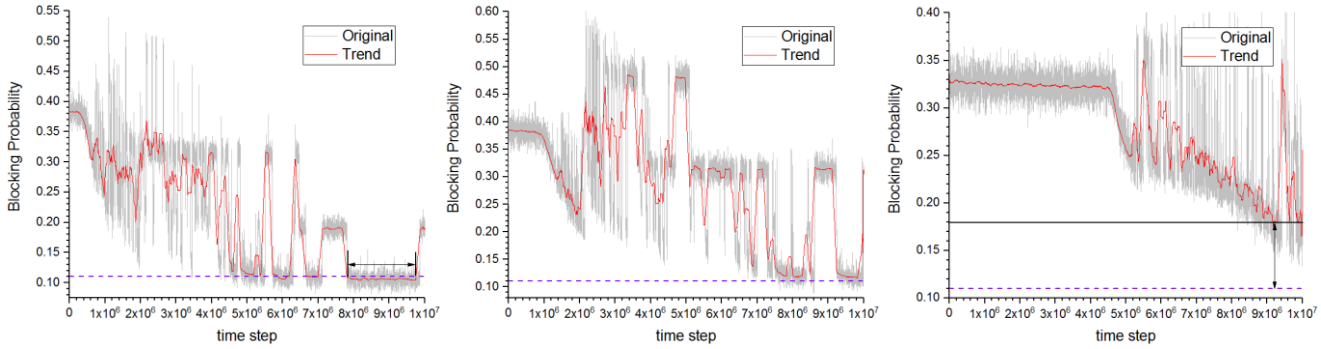
A. Learning under different topologies

There are training mode and evaluation mode while selecting the proper action. As a classifier, Actor outputs the probabilities of each action. In training mode, every action is taken by following the probability, in order to explore more possibilities. And in evaluation mode, the action with maximum probability would be taken. Fig. 4 shows the learning curves of ACRA under training mode by using both original line in gray and trend line in red. Original curves show detailed performance for every 1280 steps, trend curves show relatively smooth average performance of contiguous 12800 steps. The purple dashed line of each subgraph indicates the average blocking probability of 11% in DSP-FF. The scope indicated by the black horizontal line with a two-way arrow in Fig. 4(a) is the phase that ACRA in training mode performs better than the benchmark. In Fig. 4(a), performance of ACRA exceeds DSP-FF at about 7.8 million steps. And after about 9.7 million steps, performance of ACRA degrades, that means the training model of ACRA is



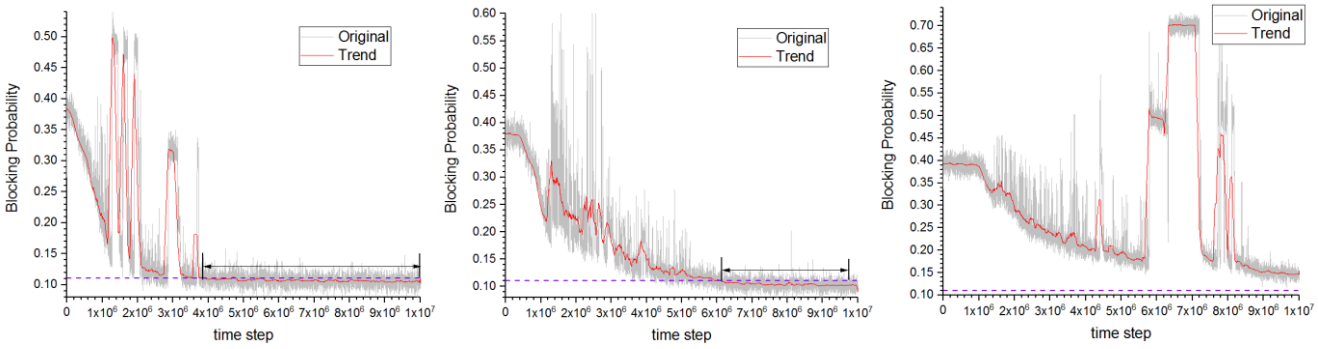
(a) (b)

Fig. 3 Alternative topologies for simulation: (a) 7 nodes and 10 links; (b) 9 nodes and 14 links.



(a) (b) (c)

Fig. 4 Learning curve with E=1 under two basic modalities on (a) 7-node and 10-link topology; (b) 9-node and 14-link topology; (c) NSFNET with 14-node and 22-link.



(a) (b) (c)

Fig. 5 Learning curve with **E=4** under **two** basic modalities on (a) 7-node and 10-link topology; (b) 9-node and 14-link topology; (c) NSFNET with 14-node and 22-link.

not stable within 10 million steps. The maximum, minimum, and average blocking probabilities in $[7.8\times10^6, 9.7\times10^7]$ are 11.541%, 10.352%, and 10.601%. So, ACRA could reduce 0.399% of blocking probability. The scope indicated by the black vertical line with a two-way arrow in Fig. 4(c) is a mark that shows the performance gap between ACRA in 10 million steps and the benchmark. The minimum value of original curve is 13.359%, that is 2.359% higher than DSP-FF. In summary, Fig. 4(a) shows that ACRA is able to learn a proper resource allocation policy better than DSP-FF in 10 million steps. But with the increase of topology complexity, learning difficulty increases either. RA problem is beyond the learning capacity of ESNet with $E=1$.

B. *Learning under different E*

Fig. 5 shows the learning curves with $E=4$. In Fig. 5(a) and (b), ACRA is able to learn a better policy than DSP-FF. And the learning process has a stable falling tendency. In Fig. 5(a), performance of ACRA becomes quite stable, that exceeds DSP-FF at about 4×10^6 step. The maximum, minimum, and average blocking probabilities in $[4\times10^6, 1\times10^7]$ are 11.276%, 10.157%, and 10.691%. So, ACRA could be relatively stable after 4 million steps, and reduce 0.309% of blocking probability with *E=4* on 7-node 10-link topology. Similarly, in Fig. 5(b), ACRA is also stable after 6.2 million steps, and could reduce 0.67% of blocking probability. The advantage of ACRA is small because the number of max-available wavelengths is much less than regular 40/80 in real WDM network, and the traffic of simulation condition of 11% average blocking probability is too heavy. In Fig. 5(c), like Fig. 4(c), ACRA is unable to perform better, although it truly learn something in simulation, that means the topology beyond NSFNET is too complex for ACRA with *E=4*. The comparison between Fig. 4 and Fig. 5 shows that in different-scale network topologies, ESNet with larger E always learning faster and better.

C. *Learning under different modalities*

Basic topology modality contains the routing information implicitly, and basic routing modality includes routing information obviously. Fig. 6 shows the learning results without basic routing modality in simulation under three topologies. In two basic modalities at Fig. 5, performance of ACRA could approach the better performance of DSP-FF on 10-link topology and 14-link topology. However, in the condition with only basic topology modality, although ACRA truly learns a better decision making as the time step increases, it could not reach the similar performance of DSP-FF. And the performance gap becomes bigger with the growth of network scale. The minimum blocking probabilities of original curves in Fig. 6(a-c) are 15.313%, 19.531%, and 23.906% respectively. Therefore, less modalities weakens the learning efficiency, and the complexity of topology still has an impact on the performance.

VI. CONCLUSION

In this paper, the concept of multi-modal optical network is introduced to control the representation for optical networks. Then a RL resource allocation algorithm is proposed, named actor-critic-based RA (ACRA) algorithm. The ACRA algorithm executes self-learning and self-resource-allocation by learning essential features from multi-

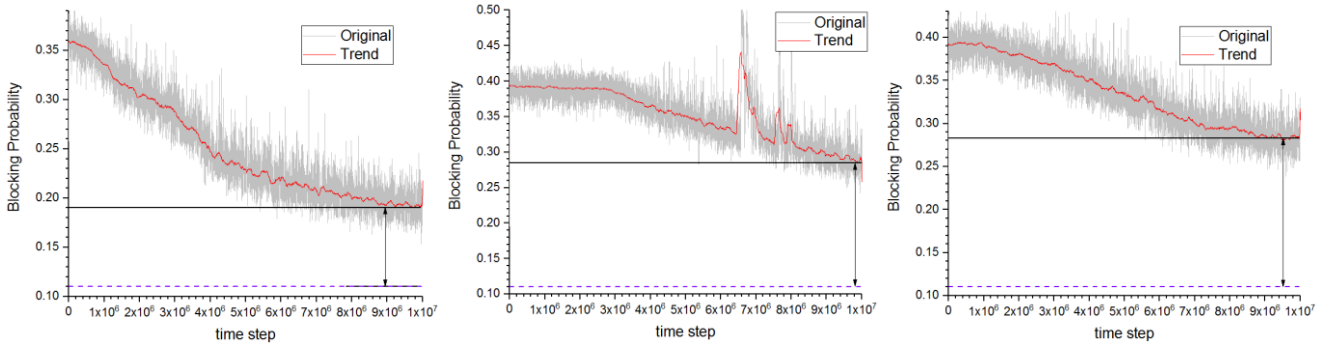


Fig. 6 Learning curve with **E=4** under basic topology modality on (a) 7-node and 10-link topology; (b) 9-node and 14-link topology; (c) NSFNET with 14-node and 22-link.

modal data. The simulation results show that ACRA is able to find out potential features and make better decision compared to traditional heuristic algorithm. We also evaluate that performance of learning task has positive correlation with width of ESNet and number of modality, and negative correlation with scale of network topology.

ACKNOWLEDGMENT

This work is supported by China State Grid Corp Science and Technology Project (Grant id: 5210ED180047).

REFERENCES

[1] "Cisco Visual Networking Index: Forecast and Methodology, 2016–2021", Cisco, [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html, [Accessed: June 2017]

[2] A. Mokhtar and M. Azizo lu, "Adaptive wavelength routing in alloptical networks," in *IEEE/ACM Trans. Netw. (TON)*, vol. 6, no. 2, pp. 197-206, Apr. 1998.

[3] J. He, M. Brandt-Pearce, Y. Pointurier, and S. Subramaniam, "QoTaware routing in impairment-constrained optical networks," in *IEEE Global Communications*, Washington, DC, USA, pp. 2269-2274, 2007.

[4] B. Ramamurthy, D. Datta, H. Feng, J. Heritage, and B. Mukherjee, "Impact of transmission impairments on the teletraffic performance of wavelength-routed optical networks," in *Journal of Lightwave Technology*, vol. 17, no. 10, pp. 1713-1723, Oct. 1999.

[5] R. Cardillo, V. Curri, and M. Mellia, "Considering transmission impairments in wavelength routed networks," in *ONDM*, Milan, Italy, pp. 421–429, 2005.

[6] A. Askarian, Y. Zhai, S. Subramaniam, Y. Pointurier, and M. BrandtPearce, "QoT-aware RWA algorithms for fast failure recovery in alloptical networks," in *IEEE/OSA OFC*, San Diego, CA, USA, pp. 1-3, 2008.

[7] G. Markidis, S. Sygletos, A. Tzanakaki, and I. Tomkos, "Impairment aware based routing and wavelength assignment in transparent long haul networks," in *Optical Network Design and Modelling*, Athens, Greece, pp. 48-57, 2007.

[8] A. Jirattigalachote, P. Monti, L. Wosinska, K. Katrinis, and A. Tzanakaki, "ICBR-Diff: an Impairment Constraint Based Routing Strategy with Quality of Signal Differentiation," in *Journal of Networks*, vol. 5, no. 11, pp. 1279-1289, Nov. 2010.

[9] N. Zulkifli and K. Guild, "Moving towards upgradeable all-optical networks through impairment-aware RWA algorithms," in *IEEE/OSA OFC/NFOEC*, Anaheim, CA, 2007.

[10] P. H. Ho, "State-of-the-art progress in developing survivable routing schemes in mesh WDM networks," in *IEEE Communications Surveys & Tutorials*, vol. 6, no. 4, pp. 2-16, Fourth Quarter 2004.

[11] Z. Zhou, T. Lin, K. Thulasiraman and G. Xue, "Novel Survivable Logical Topology Routing by Logical Protecting Spanning Trees in IP-Over-WDM Networks," in *IEEE/ACM Transactions on Networking*, vol. 25, no. 3, pp. 1673-1685, June 2017.

[12] R. S. Sutton, and A. G. Barto, Reinforcement Learning: An Introduction. The MIT Press, 2011.

[13] J. N. Tsitsiklis, "Asynchronous stochastic approximation and Q-learning," in *Proceedings of 32nd IEEE Conference on Decision and Control*, San Antonio, TX, pp. 395-400 vol.1, 1993.

[14] R.J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," in *Machine Learning*, vol. 8, no. 3, pp. 229–256, 1992.

[15] D. Silver, A Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T Graepel, and D. Hassabis, "Mastering the game of Go with deep neural networks and tree search," in *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[16] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. Van Den Driessche, T. Graepel, and D. Hassabis, "Mastering the game of go without human knowledge," in *Nature*, vol. 550, no. 7676, pp. 354-359, 2017.

[17] N. Srivastava, and R. Salakhutdinov, "Multimodal learning with deep Boltzman machines," in *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 2949-2980, 2014.

[18] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam and P. Vandergheynst, "Geometric Deep Learning: Going beyond Euclidean data," in *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18-42, July 2017.

[19] V. Mnih, A. Puigdomenech Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. "Asynchronous Methods for Deep Reinforcement Learning". in *ArXiv* preprint arXiv:1602.01783, 2016.

[20] T. Degris, P. M. Pilarski and R. S. Sutton, "Model-Free reinforcement learning with continuous action in practice," in *American Control Conference (ACC)*, Montreal, QC, pp. 2177-2182, 2012.

[21] A. Krizhevsky, I. Sutskever, G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *International Conference on Neural Information Processing Systems*, vol. 60, pp. 1097-1105, 2012.

[22] "OpenAI Baselines: ACKTR and A2C", OpenAI, [Online]. Available: https://blog.openai.com/baselines-acktr-a2c/, [Accessed: Aug. 2017].

[23] M.D. Zeiler, "ADADELTA: an adaptive learning rate method." in *arXiv* preprint arXiv:1212.5701, 2012.