DeepCMS³: A Deep Reinforcement Learning Framework for Core, Mode and Spectrum Sequential Scheduling over Optical Transport Network

Cen WANG(1), Noboru Yoshikane(1), Filippos Balasis(1), Takehiro Tsuritani(1)

(1) Photonic Transport Network Laboratory, KDDI Research Inc, ce-wang@kddi-research.jp

Abstract This paper proposes a deep reinforcement learning based framework for core, mode and spectrum resource sequential scheduling (CMS³), DeepCMS³. Compared to previous CMS³ algorithm, DeepCMS³ can achieve higher resource utilization and lower blocking rate.

Introduction

Resource optimization with objectives of higher resource utilization and lower service blocking rate is an essential part in reducing the operating expense (OPEX) over optical transport network (OTN). Classic resource optimization in a wavelength division multiplexing (WDM) based OTN is described as the routing, modulation and spectrum assignment (RMSA) problem. In RMSA, resources with only one dimension (i.e., the spectrum) need to be considered. However, the fast growth of OTN traffic has led to intensive research of space-division multiplexing (SDM), including mode-division multiplexing (MDM), a technology that can provide ultra-high bandwidth. With SDM, through the core, mode and spectrum availability it is possible to increase the capacity of OTNs in the near future. As a result, the problem's dimensions expand to three, and such three-dimensional allocation problem has been proved to be NP-hard[1].

The main challenge in getting an optimal solution for the three-dimensional resource scheduling is the huge allocation decision space. In our previous work[2] we proposed an algorithm of core, mode and spectrum sequential scheduling (CMS³), and the results are sub-optimal due to the heuristic strategy. Recently, deep reinforcement learning (DRL) has shown better performance in network resource scheduling. DRL uses deep neural networks (DNNs) to parameterize the allocation decision space, which makes it possible to fast converge to a near-optimal solution. Chen et al.[3] proposed DRL-based DeepRSMA over the elastic optical networks (EONs). Suárez-Varela et al.[4] then gave a DRL-based routing solution over the OTN.

Motivated by the related works, in this paper, we propose a DRL framework for core, mode and spectrum sequential scheduling, DeepCMS³ for short. The learning results verify that DeepCMS³ outperforms CMS³ in the resource utilization and the blocking rate.

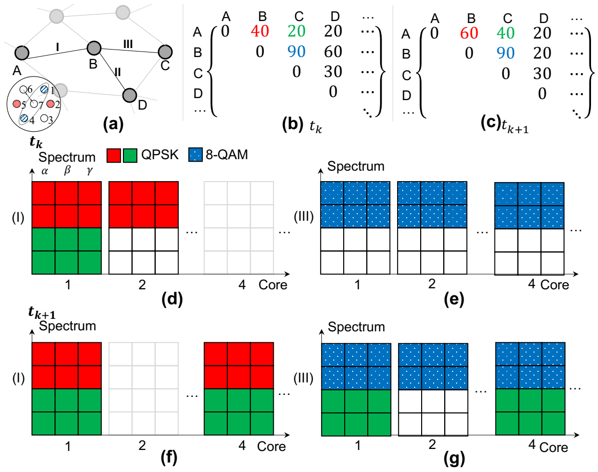

Fig. 1: Problem descriptions of the DeepCMS³.

Modeling of CMS³ and DeepCMS³

As shown in Fig. 1(a), we explain DeepCMS³ problem based on a coupled 7-core fiber, and network topology is depicted as an example. Fig. 1(b) and Fig. 1(c) are request matrices in time sequential order. Fig. 1(d) to Fig. 1(g) are parts of the specific allocation of time step t_k and t_{k+1}, where the horizontal axis is the core ID, and the vertical axis is the spectral slot. In each spectral slot, there are three coupled mode groups, in which α, β and γ represent the mode group of LP01, (LP11a, LP11b) and (LP21a, LP21b) respectively. Small rectangles with different colors and different patterns denote the allocations for different requests with different modulation formats (MF).

The crosstalk (XT) constrains are also considered in DeepCMS³ problem. The average XT is related to the core, mode and spectrum. A strongly coupled mode group with multi-input multi-output (MIMO) technology can cancel the mode related XT. Thus, the considered XT for DeepCMS³ is wavelength-dependent and can be calculated according to F. Ye, et al.[5] In addition, we should consider the XT threshold for different MFs in line with M. Klinkowski, et al.[6]. For these reasons, in Fig. 1(d), at time t_k, because the XT of AB(I) link is higher than the XT threshold for 8-QAM, the green and red requests should be modulated as QPSK, and the blue request can be transmitted as 8-QAM in BC(III) link due to acceptable XT in Fig. 1(e). In Fig. 1(f) and Fig. 1(g), at time t_{k+1}, since the wavelength-dependent XT is higher due to the longer distance (i.e., the path is AB+BC), the green request can only be allocated at the same spectrum slots in non-adjacent cores or the different spectrum slots in adjacent cores, because the XT from the allocations of the same spectrum slots in the adjacent cores is not acceptable, even if the MF is at the lowest level (i.e., QPSK in this paper).

In light of these constraints, when doing the resource allocations, the scheduling order of the requests becomes essential[7]. Because a former allocation is expected not to impact the XT satisfaction of a subsequential allocation. For example, in Fig. 1(f) if the AB request is first to be scheduled, the AB request may occupy all the spectrum slots in core 1. Then the AC request has not enough spectrum slots in two non-adjacent cores (i.e., core 1 and core 4). Although the AC request can be allocated in all spectrum slots in core 4, all the spectrum slots within cores that are adjacent to core 4 are unavailable. As a result, the AC request is preferred to be scheduled last. Given an optimal series of scheduling orders, even if the current request occupies the resources preemptively, the following request can still get enough available resources. This effect makes the specific core, specific mode and specific spectrum no longer important, and we can do the allocations preemptively. In addition, the accurate allocation of core, mode and spectrum leads to extremely huge action space of DRL, which may have a negative impact on the efficiency. In a heuristic method like previous CMS^3, the optimal series of scheduling orders is hard to be planned. Therefore, in DeepCMS^3, DRL is expected to learn the proper scheduling orders, and DeepCMS^3 is designed as follows.

Objective. To maximize the resource utilization and at the same time minimize the blocking rate are the objectives of DeepCMS^3.

State space. The state of the framework includes the traffic request matrix and the allocation of core, mode and spectrum at current time. We can use a $N^2 - N$ vector to represent the traffic request matrix. The value of the n-th (where $n = (i-1) \times N + j, i \neq j$) element in this vector denotes a x Gbps requests from node i to node j. Assuming a fiber link has C cores, M mode groups and S spectrum slots, the currently allocated resources can be represented as a $E \times (C \times M \times S)$ matrix, where E is the number of the fiber links over the OTN.

Action space. According to the earlier analysis, and following the trick from Mao et al.[7], the action can be given as $a_n = o$, where o is from $\{1,2,\ldots,N^2-N\}$ denotes the scheduling order of the n-th request. After the scheduling orders are learned, we can use the same method with CMS^3[2] but different scheduling orders to calculate resource utilization and blocking rate.

Rewards. The reward r_t is set as two parts, $r_t = U_t - R_t$, where U_t and are the resource utilization and blocking rate of timestep t respectively. Because we expect a lower blocking rate, there is a minus in front of R_t.

The framework of DeepCMS^3 is shown in the Fig. 2. The sequential requests are firstly collected by the *Network Controller* (process 1). The *Network Controller* sends topology, resource capacity and sequential traffic requests into the *OTN Simulator* (process 2). The *OTN Simulator* is working as the environment of the *DRL Agent*. The *OTN Simulator* retrieves a series of learned actions from *DRL Agent* and calculate the rewards for the *DRL Agent*. The rewards would help the *DRL Agent* to update better actions. After training, the DNNs in the *DRL Agent* are restored as the *Policy Generator* to be reused (process 5). The *Network Controller* then feeds the residual resources and traffic requests into the *Policy Generator* (process 4), the expected allocation policy is obtained then by the *Network Controller* (process 5), and the resource provisions are done accordingly (process 6).

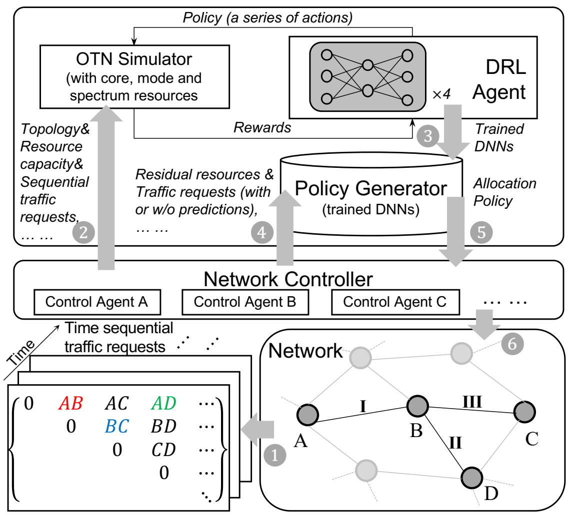

Fig. 2: DeepCMS^3 framework.

Learning and Results

We evaluate the performance of DeepCMS^3 over a Japanese transport network with 12 nodes (i.e., JPN12)[8]. The fibers in JPN12 are assumed to be step-index multicore fibers (SI-MCFs). The core radius $a_1 = 4.5\mu\mathrm{m}$, and the relative refractive-index difference is $\Delta_1 = 0.35\%$. In each fiber link, $C = 7$, $M = 3$ and $S = 40$. Each spectrum slot is 25GHz.

The traffic load is set according to the total capacity of resources over JPN12. The traffic load in all the learning processes is set as 0.6. The traffic request between each node pair is random, but the total bitrate of all the requests is determined by the load. The duration of the request is also random. The traffic matrix will be updated at each timestep. The total timestep is 100, and the training epoch is 500. DDPG[9] is used in the *DRL Agent*. As for the actor networks[9], the 3-layer CNN is used to reduce the dimension of the state representations into a vector. Then the vector is input into multiple full connected layers, and the action output is a 132 (i.e., $N^2 - N, N = 12$) vector. As for the critic networks[9], the vector of the state representations from the 3-layer CNN and the action vector are combined into a new longer vector as the input of the multiple full connected layers. The multiple full connected layers are the same in actor networks and critic networks in a learning process. There are two types of the multiple full connected layers can be chosen, one is with 6 hidden layers and the hidden size is 264 (Type 1), the other is with 3 hidden layers and the hidden size is 132 (Type 2).

The learning rewards of every 10 epochs are drawn in Fig. 3. It can be seen that DRL can converge after about 100 epochs. The neural network learning performance of Type 1 and Type 2 are also compared. The Type 2 outperforms Type 1 in getting higher rewards, but the difference is not significant.

Then, the average resource utilizations and average blocking rates of DeepCMS^3, CMS^3 and spectrum first (SF)[2] are compared in Fig. 4(a) and Fig. 4(b) respectively. The blue lines and the black dotted lines record the resource utilization and blocking rate of every 10 epochs under DeepCMS^3 with Type 1 and Type 2 respectively. The red lines and the red dotted lines are results from CMS^3 and SF respectively when the network load is also 0.6. SF strategy is that requests are allocated with spectrum resources first. It can be seen that DeepCMS^3 outperforms CMS^3 and SF in higher resource utilization and lower blocking rate due to the optimal scheduling orders.

Conclusions

We proposed DeepCMS^3, a DRL-based framework for the core, mode and spectrum sequential scheduling over the OTN. In DeepCMS^3 framework, CNN+DNN based DDPG has been proposed to reduce the dimensions of the state representations. The higher performance of DeepCMS^3 has been verified comparing to previous CMS^3 and SF algorithms.

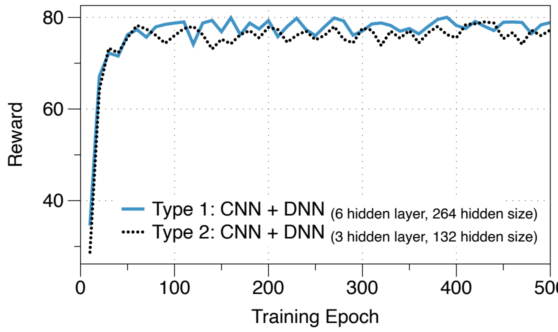

Fig. 3: Learning reward every 10 epochs.

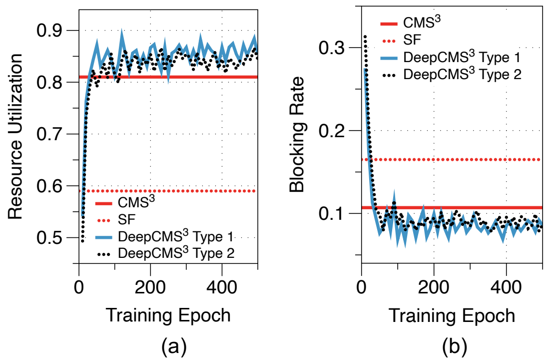

Fig. 4: The resource utilization and the blocking rate (every 10 epochs for DeepCMS^3)

References

[1] H. Hu, *et al.*, “Solving a New 3D Bin Packing Problem with Deep Reinforcement Learning Method,” *arXiv:1708.05930*, 2017.

[2] C. WANG, *et al.*, “Guaranteed-QoT, High-Availability and High- Utilization Core, Mode and Spectrum Sequential Scheduling in Optical Transport Network,” *2020 25th OptoElectronics and Communications Conference (OECC)*, Taipei, 2020.

[3] X. Chen, et al., "DeepRMSA: A Deep Reinforcement Learning Framework for Routing, Modulation and Spectrum Assignment in Elastic Optical Networks," *in Journal of Lightwave Technology*, vol. 37, no. 16, pp. 4155-4163, 15 Aug.15, 2019.

[4] J. Suárez-Varela et al., "Routing Based on Deep Reinforcement Learning in Optical Transport Networks," *2019 Optical Fiber Communications Conference and Exhibition (OFC)*, San Diego, CA, USA, 2019, pp. 1-3.

[5] F. Ye et al., "Wavelength-Dependence of Inter-Core Crosstalk in Homogeneous Multi-Core Fibers," *in IEEE PTL*, vol. 28, no. 1, pp. 27-30, 2016.

[6] M. Klinkowski, et al., "A Study on the Impact of Inter-Core Crosstalk on SDM Network Performance," *2018 ICNC*, Maui, HI, 2018, pp. 404-40.

[7] Hongzi Mao, et al., “Resource Management with Deep Reinforcement Learning,” *in Proceedings of HotNets '16, New York*, USA, 2016, pp.50–56.

[8] S. Arakawa, et al., "Topological Characteristic of Japan Photonic Network Model," *in IEICE Technical Report*, PN2013-2, Jun. 2013.

[9] Timothy P. Lillicrap, et al., “Continuous Control with Deep Reinforcement Learning,” *arXiv:1509.02971*, 2016.