CHALMERS
UNIVERSITY OF TECHNOLOGY

DRL-based RMSCA for SDM Networks with Core Switching in Multi-Core Fibres

DRL-based RMSCA for SDM Networks with Core Switching in Multi-Core Fibres

Yiran Teng*, Ruizhi Yang*, Carlos Natalino†, Sen Shen*, Paolo Monti†,
Reza Nejabati*, Shuangyi Yan*, Dimitra Simeonidou*
* High Performance Network Group, University of Bristol, Bristol, United Kingdom.
Email: {ab20471@bristol.ac.uk}
† Electrical Engineering Department, Chalmers University of Technology, Gothenburg, Sweden.

***Abstract*—We designed a DRL-based routing, modulation, spectrum, and core assignment algorithm to fully exploit the core switching capabilities of space division multiplexing networks with multi-core fibres. The proposed method shows up to 53% better blocking probability performance compared to RMSCA benchmarks in different traffic loads.**

***Index Terms*—Deep Reinforcement Learning, Space Division Multiplexing, Multicore Fibres, Elastic Optical Networks**

I. Introduction

Recent advancements in transmitting devices that support multi-core fibres (MCFs) have made space division multiplexing (SDM) a promising technology in optical networks to support an ever-growing network traffic. The deployment of MCFs requires a solution to the routing, modulation, spectrum, and core allocation (RMSCA) problem able to: *(i)* efficiently handle fibre cores and frequency slots (FSs) and *(ii)* maximise network performance (e.g., number of accepted connections).

Previous studies on RMSCA algorithms focused on heuristic methods [1]–[3] under the constraint of the inter-core crosstalk (XT). However, heuristic algorithms may lead to far-from-optimal solutions due to their inability to assess the impact of current RMSCA decisions on the provisioning results of future connection requests. Recently, deep reinforcement learning (DRL) showed promising results in solving the dynamic connection provisioning/reconfiguration problem in single-core elastic optical networks (EONs) [4]–[7]. In these works, DRL employs deep neural networks (DNNs) to extract network state information and optimise a long-term cumulative reward. In MCF-based SDM networks, the DRL-RMSCA algorithm [8] presented by *Pinto-Ríos et al.* selects only aligned cores, failing to exploit the full core switching capability of MCF-EON. On the other hand, when developing a DRL-based RMSCA scheme able to switch between cores, handling the cardinality of the observation and the action spaces becomes challenging. Additionally, with an extremely large action space, it might not be possible to guarantee that invalid actions are not ignored by a trained DRL agent, which negatively impacts the blocking probability (BP), or will require a higher training cost [8].

This paper develops and validates a DRL-based RMSCA framework for MCF-EONs with core switching capabilities, namely DRL-RMSCA-CS. The framework is composed of a DRL agent responsible for jointly solving the routing and core allocation problems. A detailed representation that integrates crucial network state information (e.g., usage and fragmentation) is used to help the DRL agent to better perceive the network condition at a given point in time. Additionally, an action mask is applied to help the DRL agent to avoid selecting invalid actions. Finally, a fragmentation-aware reward function is used to guide the agent in maximising resource usage, thus lowering the overall network blocking probability. Simulation results over the NSFNET network show how DRL-RMSCA-CS is able to reduce blocking probability by up to 53.6% when compared to two state-of-the-art RMSCA heuristics. In conclusion, DRL-RMSCA-CS is an innovative and effective DRL-based solution for the RMSCA problem in small-scale MCF-EONs (NSFNET with three cores).

II. RMSCA Problem with Core Switching

The topology of MCF-EONs is modelled as a graph $G(V, E, C)$, where V, E, C represent the set of nodes, links, and cores, respectively. F FSs with two states (0 when free, 1 when occupied) are located on each core. The spectrum state of the MCF-EONs is represented by a $E \times C \times F$ matrix denoted as $FSM \in [0, 1]$.

We assume nodes can perform core switching, so the RMSCA algorithm has the possibility to select the core to be used at each link in the path. The list of core permutations (CPs) represents the core-id options to be used on each link of the selected path. When taking core switching into account, the total number of CPs is $|C|^n$ for a path with n hops. Upon the arrival of a request $R_t(s, d, b)$, an RMSCA algorithm selects a CP on one of the paths from nodes s to d and assigns enough continuous and contiguous FSs to meet the bandwidth b.

III. The DRL-RMSCA-SC Framework

Our framework is composed of four elements: the *MCF-EON* environment, the *Preprocessor*, the *DRL Agent*, and the *Trainer*. Fig. 1 illustrates the interaction among them. Specifically, when R_t arrives at timestep t, the Feature Extractor receives the filtered information from the Preprocessor (step 2) and generates the state vector s_t (step 3). Then, s_t is fed into the DNNs, which outputs an action a_t. The RMSCA Policy takes a_t and executes it on the MCF-EON (step 4). Subsequently, the Evaluator computes the reward r_t using the feedback from the RMSCA action (step 5). The tuple (s_t, a_t, r_t) is stored in the Experience Buffer as a training

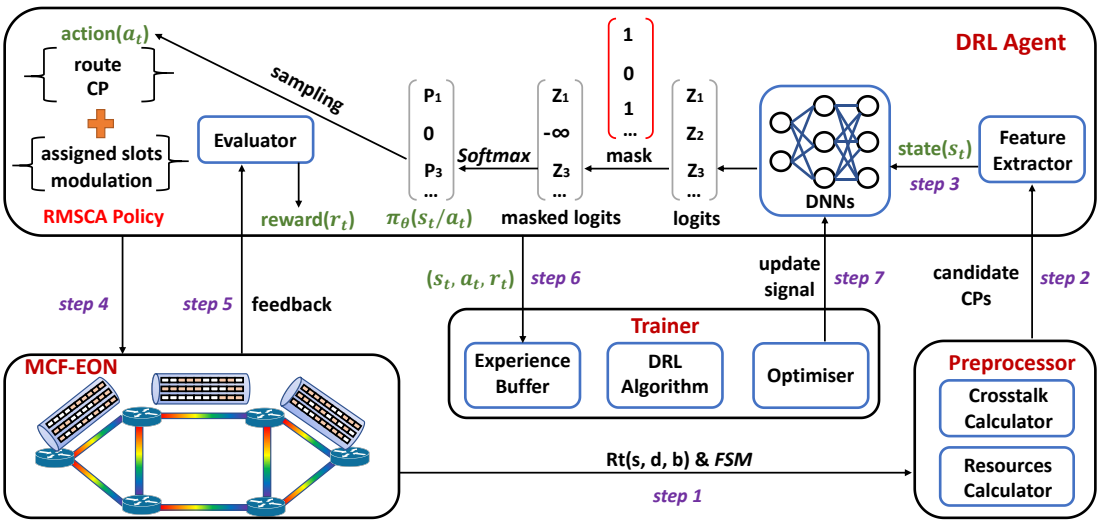

Fig. 1. The DRL-RMSCA-SC framework.

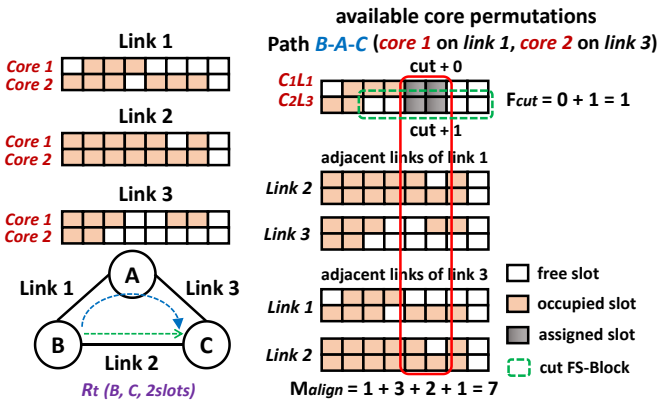

Fig. 2. The calculation of F_{cut} and M_{align}.

sample (step 6). The training process is triggered when the Experience Buffer is full. At this step, the Optimiser is used to update the parameters of the DNNs (step 7).

1. State representation: to reduce the action space and mitigate the inter-core XT, a Preprocessor selects candidate CPs. First, all CPs with enough FSs are identified using the information in the FSM. Then, M candidate CPs with the lowest overlapping slots in neighbouring cores are selected for each one of the precomputed k shortest paths between s and d.

Then, state s_t is designed based on the request information and the spectrum state of the candidate CPs. s_t is a $(2N + 7MK)$ vector, where $2N$ elements in one-hot format represent s and d. For each of the $M \times K$ candidate CPs, 7 essential features are included: *(i)* number of required FSs, *(ii)* number of hops, *(iii)* number of adjacent links, *(iv)* number of available slots, *(v)* start index of the first available FS block, *(vi)* length of the first available FS block, and *(vii)* average number of free slots located at the same position as the first available FS-block on the adjacent links of each core.

2. Action space: the action space a_t is a $M \times k + 1$ vector. Each action represents either selecting one of the M CPs on one of the k paths, or rejecting the request. The spectrum is allocated based on the first-fit method. The modulation scheme is determined by the physical length of the path and the inter-core XT constraints [9], [10]. To prevent the agent from choosing unavailable CPs, we adopt an action mask [11] (Fig. 1). The mask is a $M \times k+1$ vector $\in$ [0,1], corresponding to the availability of each action. After masking, the original logits of each invalid action are replaced by an infinitesimal number making their output probability after the $softmax$ activation function equal to 0.

3. Reward: we build a reward r_t to guide the agent towards an effective RMSCA strategy. If a request is rejected, $r_t = -1$. Otherwise, the impact of the current RMSCA action on MCF-EON is computed using cost C, computed as follows:

$$\mathrm{C} = \frac{S_o + 10 \times F_{cut} + \frac{M_{align}}{n}}{S_a}, \tag{1}$$

where n is the number of cores, S_o and S_a represent the total number of assigned slots for the request R_t and available slots on the selected CP, respectively. F_{cut} and M_{align} are related to slot fragmentation and misalignment in the network. As shown in Fig. 2, F_{cut} denotes the number of FS blocks that are cut into sub-blocks by the assigned slots, and M_{align} denotes the total number of free FSs on the adjacent links of each core of the selected CP that have overlapping spectrum. The reward r_t is designed as follows:

$$r_t = \begin{cases} 1 + \dfrac{A}{M \times k} \times 0.5, & \text{if } C = C_{min} \\ max(-0.1 \times C + 0.6,\ 0), & \text{otherwise} \end{cases}$$

where A is the number of available CPs among $M \times K$ candidate CPs. r_t is negatively correlated to the cost and a large reward is given to the agent if the selected CP leads to the lowest cost C_{min} among all candidate CPs in order to encourage the agent to learn an efficient policy.

IV. Performance Assessment

The MCF-EON environment is simulated using the Optical RL-Gym [12]. The NSFNET topology with 14 nodes and 22 links is used for evaluation. We set 3 cores on each link and 100 FSs with 12.5 GHz bandwidth on each core. Inter-core XT is ignored due to its negligible impact on transmission reach MCFs with 7 cores or less [10]. The bit rate requirement of each request ranges from 25 to 100 Gb/s. M and k are both set to 5. The proximal policy optimisation (PPO) algorithm is used for training the DRL agent. The batch size, discount factor, clip factor, and learning rate are set to 500, 0.95, 0.2, and 10^{-5}, respectively.

We compare the BP performance of our algorithm (i.e., DRL-RMSCA-CS) with two heuristics, i.e., k-shortest-path first-core first-fit with aligned cores (KSP-FC-FF-CA) and core-switching (KSP-FF-FC-CS). As shown in Fig. 3, DRL-RMSCA-CS converges after 500,000 requests to a BP value which is 51.5% and 40.1% lower than KSP-FF-FC-CA and KSP-FF-FC-CS, respectively.

Fig. 4 shows that DRL-RMSCA-CS achieves the best performance consistently across traffic loads ranging from 250 to 350 Erlang. Moreover, results show that higher gains can be achieved in lower loads. For instance, DRL-RMSCA-CS achieves 83.1% and 79.3% lower BP than KSP-FC-FF-CA and KSP-FF-FC-CS, respectively, when the load is 250 Erlang. We speculate that this is attributed to the large number of available CPs at lower loads, which increases the possibility for the agent to select good actions. When the load increases, there are less opportunities for improving the core assignment.

V. Conclusion

We investigate the RMSCA problem when core switching is possible and propose a DRL- based framework that drastically reduces the connection blocking probability in NSFNET with 3 cores. We expect the framework to present similar advantages in larger networks with more cores, but this scalability assessment is left as future work.

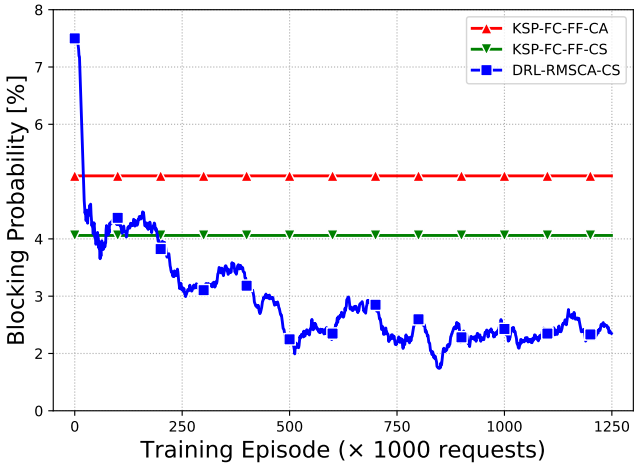

Fig. 3. Blocking probability for 300 Erlang.

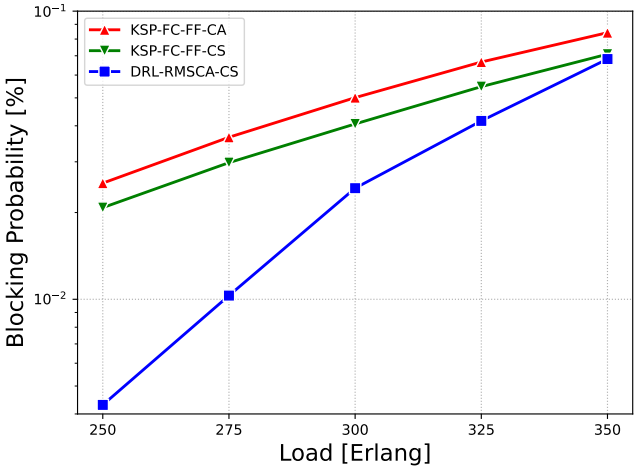

Fig. 4. Blocking probability in different traffic loads.

Acknowledgements

This work was partially supported by the CSA Catapult - UoB collaboration project, and by Sweden's innovation agency VINNOVA, within the framework of the EUREKA cluster CELTIC-NEXT project AI-NET-PROTECT (2020-03506).

References

[1] Hideki Tode et al. Routing, spectrum and core assignment for space division multiplexing elastic optical networks. In *Proc. of Networks*, 2014. DOI: 10.1109/NETWKS.2014.6958538.

[2] Piotr Lechowicz et al. Fragmentation metrics in spectrally-spatially flexible optical networks. In *Proc. of ONDM*, pages 235–247. Springer, 2020. DOI: 10.1007/978-3-030-38085-4_21.

[3] Qiuwang Lan et al. A fragmentation-aware load-balanced RMSCA algorithm in space-division multiplexing elastic optical networks. In *Proc. of ICTC*, pages 56–60, 2022. DOI: 10.1109/ICTC55111.2022.9778371.

[4] Xiaoliang Chen et al. DeepRMSA: A deep reinforcement learning framework for routing, modulation and spectrum assignment in elastic optical networks. *J Lightw. Technol.*, 37(16):4155–4163, 2019. DOI: 10.1109/JLT.2019.2923615.

[5] Bixia Tang et al. Heuristic reward design for deep reinforcement learning-based routing, modulation and spectrum assignment of elastic optical networks. *IEEE Commun. Lett.*, 26(11):2675–2679, 2022. DOI: 10.1109/LCOMM.2022.3195778.

[6] Masayuki Shimoda and Takafumi Tanaka. Mask RSA: End-to-end reinforcement learning-based routing and spectrum assignment in elastic optical networks. In *Proc. of ECOC*, 2021. DOI: 10.1109/ECOC52684.2021.9606169.

[7] Ehsan Etezadi et al. DeepDefrag: A deep reinforcement learning framework for spectrum defragmentation. In *Proc. of GLOBECOM*, pages 3694–3699, 2022. DOI: 10.1109/GLOBECOM48099.2022.10000736.

[8] Juan Pinto-Ríos et al. Resource allocation in multicore elastic optical networks: A deep reinforcement learning approach. *Complexity*, 2023. DOI: 10.1155/2023/4140594.

[9] Zuqing Zhu, Wei Lu, Liang Zhang, and Nirwan Ansari. Dynamic service provisioning in elastic optical networks with hybrid single-/multi-path routing. *J Lightw. Technol.*, 31(1):15–22, 2012. DOI: 10.1109/JLT.2012.2227683.

[10] Jordi Perelló et al. Flex-grid/SDM backbone network design with inter-core XT-limited transmission reach. *J. Opt. Commun. Netw.*, 8(8):540–552, 2016. DOI: 10.1364/JOCN.8.000540.

[11] Shengyi Huang and Santiago Ontañón. A closer look at invalid action masking in policy gradient algorithms. *Proc. of FLAIRS*, 2022. DOI: 10.32473/flairs.v35i.130584.

[12] Carlos Natalino et al. The Optical RL-Gym: An open-source toolkit for applying reinforcement learning in optical networks. In *Proc. of ICTON*, 2020. DOI: 10.1109/ICTON51198.2020.9203239.