European Conference on Optical Communication (ECOC) 2022 © Optica Publishing Group 2022

MAGC-RSA: Multi-Agent Graph Convolutional Reinforcement Learning for Distributed Routing and Spectrum Assignment in Elastic Optical Networks

Huy Tran Quang(1), Omar Houidi(2), Javier Errea-Moreno(1), Dominique Verchere(1), Djamal Zeghlache(2)

(1) NOKIA Bell Labs, France, huy.h.tran@nokia.com
(2) Telecom SudParis, Samovar-Lab, Institut Mines-Telecom, Institut Polytechnique de Paris, France

Abstract
This paper proposes MAGC-RSA, a Multi-Agent Graph Convolutional Reinforcement Learning approach, to solve the Routing and Spectrum Assignment (RSA) problem in a distributed manner. A blocking probability reduction of 80% can be achieved compared to the Shortest Path First-Fit approach. ©2022 The Author(s)

Introduction

Bandwidth demands of end-to-end 5G network slices and cloud services have significantly increased over the years. Thus, optical transport networks have evolved to better adapt to this requirement with Elastic Optical Networks (EONs) technology and Software-Defined Networking (SDN) paradigm. EONs provide the flexibility to assign more fine-grained spectral slot width with regards to the traffic requests, hence, improve notably resource utilization. In addition, SDN controller (SDNC) and open standard interfaces (i.e. OpenConfig, Transport API, etc.) enable the automation of network configuration, control and monitoring. These are key enablers for the implementation of Machine Learning (ML) and Deep Reinforcement Learning (DRL) techniques in optical networks.

ML techniques such as Neural Network, Tree-based and K-means methods are applied to solve both supervised and unsupervised learning tasks in Quality of Transmission estimation, traffic prediction and failure early detection[1]. On the other hand, the Routing and Spectrum Assignment (RSA) is one of the fundamental challenges which belongs to the set of decision-making problems in EON. The RSA algorithm computes a path between the source and destination as well as a block of appropriate frequency slots (FS) while ensure the spectrum continuity and contiguity constraints in a dynamic network scenario. Thus, deep reinforcement learning is considered a good candidate to solve this problem either in single or multi-agent (MADRL) approach such as MaskRSA[2] or DeepRMSA[3], respectively. However, the single-agent DRL approach may lead to a long training period. Moreover, relational features, such as non-fragmented spectrum block between links in a path, need to be extracted by manually applying an equivalent kernel.

In this paper, we propose the multi-agent graph convolutional reinforcement learning approach called *MAGC-RSA* to solve the Routing and Spectrum Assignment problem in a distributed manner. Leveraging the similarity of network topology and graph structure, GCN models each DRL agent as a network node in the topology. Based on the global view provided by SDNC and the observation of its neighbors, the DRL agent, which corresponds to the the source node of the request, makes the decision to select the optimal path and spectrum resources. Moreover, we adopt the Attention mechanism[4] as the kernel in the convolutional layer to extract latent features for faster convergence even in a large observation space of many nodes. Additionally, DRL agents are trained with a fragmentation-aware reward function, which leads to a better spectrum utilization. To the best of our knowledge, this centralized training and distributed execution method is applied for the first time to solve the RSA problem in EONs. Our approach achieves a lower blocking probability as compared to the heuristic K-Shortest Path First-Fit and another MADRL solution.

MAGC-RSA: Multi-Agent Graph Convolutional Reinforcement Learning for RSA problem

We model the MAGC-RSA algorithm as a graph $G = (N, E)$, where agents represent network nodes and edges express the connection between them. For every service request, the agent that corresponds to the source node is in charge of finding the optimal path and spectrum resources. Fig. 1 describes the operation principle of our proposed solution.

MAGC-RSA architecture consists of three main components: the Encoder, the Convolutional Layer and the Q Network. They are illustrated in Fig. 2 and described as follows:

1) Observation Encoder: The local observation of agent a_i at time step t is denoted as vector $o_i(t)$. Each observation vector $o_i(t)$ contains the one-hot encoding of the source and destination of the request, the number of requested frequency slots, the

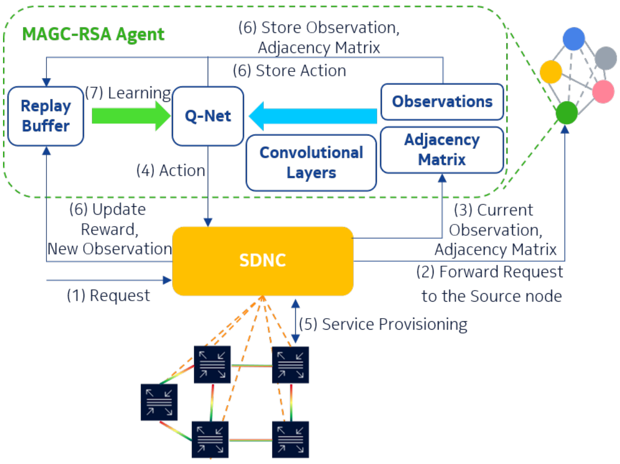

Fig. 1: Operation principle of MAGC-RSA

spectrum utilization of k candidate shortest paths between the source and destination, the spectrum utilization of d links originating from K neighbors of a_i. The value of d, K may vary between environments. In this paper, we choose $d = 3$, $K = 4$. Once constructed, each observation vector will be encoded by the Multi Layer Perceptron (MLP) to create the feature vector f_i of agent a_i.

The feature vector f_i is defined by multiplying two matrices $M_{Adj} \times F(t)$. M_{Adj} is a $N \times N$ matrix, where $m_{i,j} = 1$ if agent a_i has a connection towards a_j, $m_{i,j} = 0$ otherwise. $F(t)$ is the feature matrix that aggregates all feature vectors of all agents at time t.

2) Convolutional layers: These layers take as input the feature vector f_i, which is the output of MLP, and an adjacency matrix M_{Adj} of the agent. The convolutional layer, namely the Kernel relation, is responsible for generating the latent features. In this context, the convolutional kernel uses multi-head dot-product attention to compute interactions between agents. Considering an agent a_i and the set $\mathcal{E}_i$ of its K neighbors, an input feature is expressed in a query, key and value representation by each independent attention head[4].

In order to model the relationship between each agent and its neighboring agents (i.e., between i and $j \in \mathcal{E}_i$) for an attention head m, we use the following formula:

$$\alpha_{ij}^m = \frac{exp(\delta \ . \ W_q^m f_i \ . \ (W_k^m f_j)^T)}{\sum_{e \in \mathcal{E}_i} exp(\delta \ . \ W_q^m f_i \ . \ (W_k^m f_e)^T)} \quad (1)$$

where: for each agent a_i, there are a set of entities $\mathcal{E}_i$ (K neighbors and the node n_i itself) in the local region, and δ is a scaling factor.

For each attention head, the value representations of all the input features are weighted by the relation and summed together. Subsequently, for agent i the outputs of the M attention heads are concatenated and fed into the σ-function (a MLP with ReLU non-linearities) in order to produce the convolutional layer

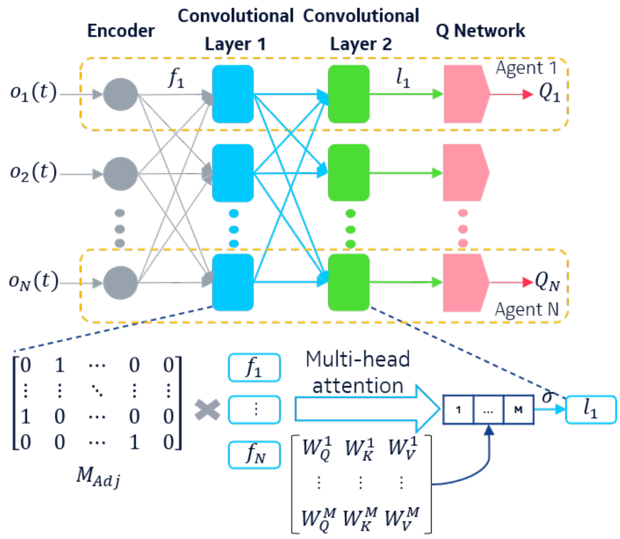

Fig. 2: MAGC-RSA architecture.

output. Therefore, the latent feature vector l_i is formulated as follows:

$$l_i = \sigma(concatenate[\sum_{j \in \mathcal{E}_i} \alpha_{ij}^m W_v^m f_j, \forall m \in M]) \quad (2)$$

3) Q-Network: For each agent, the Q-network is applied on the concatenated features of the previous layers, allowing to take into consideration the cooperation at different scopes. The Q-network aims here to set the actions by calculating the Q-values. Therefore, during the training phase, the tuple (O, A, O', R, M_{Adj}) is stored into the buffer B at each time step t; where: O= (o_1, ... o_N) is the set of observations, A= (a_1, ... a_N) is the set of actions, O'= (o'_1, ... o'_N) is the set of next observations, R= (r_1, ... r_N) is the set of rewards, and M_{Adj}= (M_{Adj_1}, ... M_{Adj_N}) is the set of adjacency matrix. Next, a random mini-batch of size S is sampled from B and hence, we minimize the following loss:

$$Loss_Q(\theta) = \frac{1}{S}\sum_S \frac{1}{N}\sum_{i=1}^{N}((r_i + \gamma max_{a'} Q(O'_i, a'_i; \theta')) - Q(O_i, a_i; \theta))^2 \quad (3)$$

With: γ is the discount factor, and θ is the parameterization of the Q function.

It is worth noting that during the computation of Q-loss in the learning phase, the underlying graph can change over time, which prevents the convergence of Q and leads to some learning instabilities. To deal with the latter issue, M_{Adj} is kept unchanged in two successive time steps. Therefore, in order to update the parameters of the latter scheme, the Q-loss gradients of all agents are accumulated. Each agent minimizes not only its own Q-loss but also the Q-loss of the other agents it collaborates with. Each agent communicates only with its K neighbors which makes the scheme easily applicable to large-scale

GC-MARL systems.

Deep-Q-learning (DQL) is implemented to train our model where the future value estimation is used as the target for the current estimation. We apply the temporal relation regularization to keep the learning attention weight distribution stable over time-steps.

Actions and Rewards

1) Actions: Given the observation vector and adjacency matrix described in the previous section, a MAGC-RSA agent takes an action based on the output of the Q-Network. It selects the path and the first index of the contiguous frequency slots block among $k \times J$ actions in the discrete action space; where k is the number of candidate shortest paths, and J is the number of sufficient frequency slots blocks.

2) Rewards: The reward function considers not only the resource allocation capability of the agent for a given request, but also providing useful information concerning the impact of this assignment with regards to the effect of fragmentation. For that reason, we introduce the Shannon entropy fragmentation metric ${H_{frag}}$[5] in the reward function as follows:

$$R = \begin{cases} 1 + e^{-H_{frag}} & \text{successfully allocated} \\ -1 & \text{otherwise} \end{cases}$$

Since large values of H_{frag} indicate higher levels of fragmentation, we convert this metric using the exponential function. The agent gets a positive reward 1 plus an additional fragmentation-aware metric if it can accommodate the request. Otherwise, it receives a negative reward -1.

Experiment Setup

We conducted the experiments by extending the environment in[6]. In our implementation, all agents cooperate and interact with one single environment. Thus, after performing an action, all agents update their observations and adjacency matrix, based on the network information provided by SDNC. The simulation considers the dynamic scenario where requests arrive following a Poisson process with arrival rate of 10 and have the mean service duration of 25 units of time, which follows the exponential distribution. Each request's source-destination pair is randomly selected and its bandwidth demand are evenly distributed between [2-4] frequency slots. The agents were trained in 1000 episodes. Each episode consists of 10000 services. The hyperparameters are specified in Table 1.

Performance Evaluation

The performance of MAGC-RSA is experimented on the 14-node NSFNET topology for small-scale problems and the 28-node Pan-European topology for larger-scale problems.[7]

Tab. 1: Hyperparameters

Parameters	Values
Learning rate	10^{-4}
Batch size	64
# neighbors	4
# attention head	4
# hidden layers	2
Hidden layers dimension	64
Discount factor γ	0.96

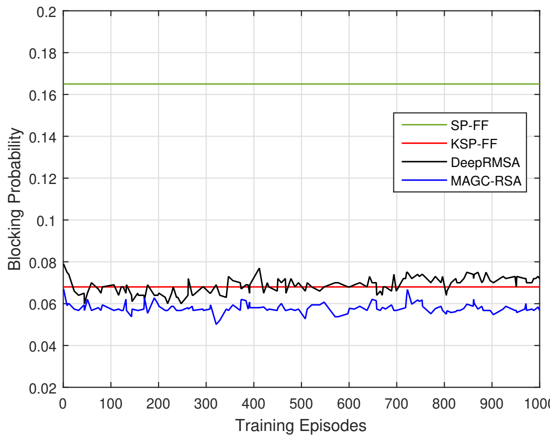

Fig. 3: Request blocking probability with NSFNET topology.

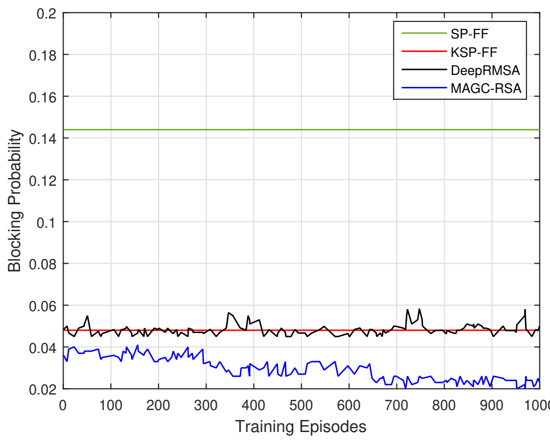

Fig. 4: Request blocking probability with Pan-European topology.

Fig. 3 and Fig. 4 depict the results in terms of request blocking probability. Our MAGC-RSA algorithm outperforms DeepRMSA, SP-FF, and KSP-FF by accepting more requests in long-term. MAGC-RSA reduces the blocking probability by 16.62% as compared to DeepRMSA.

Conclusions

This work shows that Multi-agent graph convolutional-based reinforcement learning approach is capable of improving performance in terms of blocking probability beyond traditional RSA techniques and previous machine learning-based methods. MAGC-RSA's results also show that the performance can be further improved via efficient cooperation using multi-head attention, acting as the convolutional kernel, to compute interactions between agents.

European Conference on Optical Communication (ECOC) 2022 © Optica Publishing Group 2022

References

[1] R. Gu, Z. Yang, and Y. Ji, "Machine learning for intelligent optical networks: A comprehensive survey", *CoRR*, vol. abs/2003.05290, 2020. arXiv: 2003.05290. [Online]. Available: https://arxiv.org/abs/2003.05290.

[2] M. Shimoda and T. Tanaka, "Mask rsa: End-to-end reinforcement learning-based routing and spectrum assignment in elastic optical networks", in *2021 European Conference on Optical Communication (ECOC)*, 2021, pp. 1–4. DOI: 10.1109/ECOC52684.2021.9606169.

[3] X. Chen, B. Li, R. Proietti, H. Lu, Z. Zhu, and S. J. B. Yoo, "Deeprmsa: A deep reinforcement learning framework for routing, modulation and spectrum assignment in elastic optical networks", *Journal of Lightwave Technology*, vol. 37, no. 16, pp. 4155–4163, 2019. DOI: 10.1109/JLT.2019.2923615.

[4] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, *Attention is all you need*, 2017. DOI: 10.48550/ARXIV.1706.03762. [Online]. Available: https://arxiv.org/abs/1706.03762.

[5] P. Wright, M. C. Parker, and A. Lord, "Minimum- and maximum-entropy routing and spectrum assignment for flexgrid elastic optical networking [invited]", *Journal of Optical Communications and Networking*, vol. 7, no. 1, A66–A72, 2015. DOI: 10.1364/JOCN.7.000A66.

[6] C. Natalino and P. Monti, "The optical rl-gym: An open-source toolkit for applying reinforcement learning in optical networks", in *2020 22nd International Conference on Transparent Optical Networks (ICTON)*, 2020, pp. 1–5. DOI: 10.1109/ICTON51198.2020.9203239.

[7] A. Betker, C. Gerlach, R. Hulsermann, *et al.*, "Reference transport network scenarios", *MultiTeraNet Report*, pp. 1–15, Jan. 2003.