A Reinforcement Learning Framework for Parameter Optimization in Elastic Optical Networks

Rebekka Weixer(1), Sebastian Kühl(1), Rui Manuel Morais(2), Bernhard Spinnler(3), Wolfgang Schairer(3), Bernd Sommerkorn-Krombholz(3), Stephan Pachnicke(1)

(1) Kiel University, Kaiserstr. 2, 24143 Kiel, Germany, rebekka.weixer@tf.uni-kiel.de
(2) Infinera Portugal, Rua da Garagem 1, 2790-078 Carnaxide, Portugal
(3) Infinera Germany, Sankt-Martin-Str. 76, 81541 Munich, Germany

Abstract *We present a reinforcement learning (RL) framework for maximizing the total capacity of a 51-channel transmission system, which runs magnitudes faster than a genetic algorithm (GA) based optimization. The generalization capabilities and performance of the RL framework are compared to results obtained with a GA.*

Introduction

Elastic Optical Networking (EON) provides a technology to manage the growing bandwidth demand by efficient use of spectral resources. Flexible devices, such as reconfigurable optical add/drop multiplexers (ROADM) and variable bandwidth transponders (BVT) are used for this purpose. A ROADM enables the steering of routes between different node directions whereas a BVT allows adapting the modulation format, coding scheme, forward error correction overhead and symbol rate according to the current link conditions. This enhanced operational flexibility often results in a significant increase in optimization complexity. Selecting the best set of parameters (modulation format, launch power, etc. …) from the huge parameter space is a difficult challenge, but simultaneously the key for achieving high capacities in EONs. In order to address this challenge, network planning requires an accurate estimation of the physical layer impairments (PLI), which include amplifier noise (amplified spontaneous emission, ASE) and non-linear interference (NLI)[1]. Currently, the most elaborated PLI model is based on Gaussian noise (GN) model versions[2],[3], which can be converted to fast performing closed form analytical expressions with reasonable approximation assumptions[4]. They combine adequate accuracy with relatively low computational complexity, to calculate a generalized OSNR (GOSNR) taking PLI and ASE into account. Based on this model, fixed-grid wavelength division multiplexing (WDM) networks with static traffic requests have already been optimized with the aim of maximizing the overall network throughput, by adapting the launch power and modulation format for each channel[5]. For EONs and dynamic traffic requests, an optimization of the link-level resource allocation was presented in[1] and extended to multi-point networks[6], by using a combination of the GN model with a transmission reach method[6].

Beyond the conventional algorithms, heuristic and machine learning algorithms are being investigated as promising approaches for resource allocation optimization. In[7] an adaptive GA for solving dynamic routing, modulation and spectrum assignment (RMSA) for EONs is proposed, which is designed for multi-objective optimization. Another approach is presented in[8], in which a RL algorithm is applied.

In this paper, we present a RL framework to optimize the process of setting up controllable BVT parameters with the aim of maximizing the total capacity of a point-to-point link. During the training process, the RL agent interacts with the GN model to learn the complex nonlinear dependencies of the transmission system. Subsequently, it can apply the acquired knowledge to adjust the BVT parameters depending on the current conditions in a short time (i.e. in the order of seconds). Results from a GA[9] are taken to benchmark the obtained results.

Reinforcement Learning

In RL, an agent learns by repeatedly interacting with an environment over a number of discrete time steps, t. Based on the observation of the environment, the agent selects an action (from a range of possible actions). A reward is then attributed to the agent based on the outcome of the performed action[10]. The resulting behaviour is called the policy of the agent, which can be understood as a mapping between an observation state of the environment and an action to perform. The goal of RL is then to develop a generalized policy that maximizes the long-term expected reward.

The optimization problem under study can be modeled as a Markov decision process (MDP), which makes RL algorithms suitable for the task[10]. A MDP is defined as a tuple (S, A, R, T): where S is the set of possible states of the

environment, A is the set of actions the agent can perfom, $R: S \times A \times S \to \mathbb{R}$ is the reward function for the agent which depends on the current state, the action performed and the next state, and $T: S \times A \times S \to [0,1]$ is the transition function, where $T(s_t, a_t, s_{t+1}) = \Pr(s_{t+1}|s_t, a_t)$ is the probability distribution of the next state s_{t+1} given the current state s_t and action a_t for a time step t. A policy π specifies which action should be executed when presented with a certain state. The goal of training an agent is to find an optimal policy $\pi^*: S \mapsto A$ that maximizes the long term reward. The optimal policy can be found as[11]:

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\tau\sim\pi(\tau)}[\tau] = \arg\max_{\pi} \mathbb{E}_{\tau\sim\pi(\tau)}[\sum_{t=1}^{T} R(s_t, a_t)] \quad (1)$$

where the expectation, $\mathbb{E}$, is taken over $\sum_{t=1}^{T} R(s_t, a_t)$, τ is a trajectory defined by a sequence of states and actions, which define a single episode of length T. In order to find π^* a balance between exploitation and exploration is required. By exploring the state space the agent can gather useful experience about possible states and rewards, whereas by exploiting acquired knowledge the agent will select the best action given its current experience. A common method to balance exploration/exploitation is the so-called ϵ-greedy approach. Under this method the agent selects a random action with probability ϵ, otherwise the action is selected using the current policy. During the training phase ϵ is often decreased continously.

Simulation Setup

The simulation setup (see Fig. 1) consists of the RL agent and the optical transmission system which defines the environment. A set of N_c coherent channels, $\boldsymbol{c} = [c_1, \ldots, c_{N_c}]$, are to be transmitted over a WDM link using a fixed channel spacing. It is assumed that the transmission is done over N_s spans. Each channel c_i is defined by a central frequency f_i, power p_i, bit rate b_i and modulation format mf_i.

The PLIs are then estimated using the GN model[2],[3], which provides the GOSNR_i and the GOSNR margin for each channel by:

$$m_i = \mathrm{GOSNR}_i - \text{required } \mathrm{OSNR}_i. \quad (2)$$

The GN model is implemented on the assumption that all spans and amplifiers are identical. Furthermore, p_i at each amplifier is the same, thus no amplifier gain or fiber attenuation tilt or gain ripple are considered.

The agent's goal is to maximize the overall bit rate while maintaining a predefined minimum m_i. To solve the problem by means of RL it is required to define the environmental properties in states, actions and rewards (see Fig. 1).

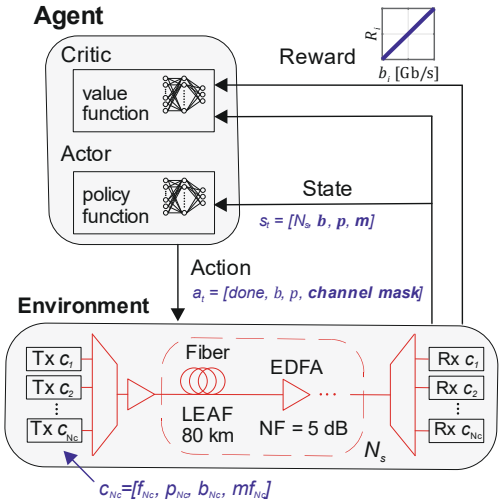

Fig. 1: Simulation setup.

State: s_t is a vector of size $(1 + 3N_c)$, containing the information about the number of spans of the system, and the bitrate, power, and margins of each channel.

Action: the agent is able to set a new bitrate and power for a selected number of channels. The ones to be changed are addressed with a channel mask $\boldsymbol{\xi} \in [-1,1]$ of length N_c. Note that $\xi_i \geq 0$ indicates that c_i will be affected. Additionally, the agent is able to signal that no further changes to the current parameters are necessary by setting a *done* flag. Summarizing, the agent sets a bitrate and power for the masked subset of channels. Thus, the action space representation a_t is a vector of size $(3 + N_c)$.

Reward: after each action the agent receives a small negative reward of $-1\,/\,(2N_c + 1)$ to encourage the agent to find the best possible solution with the smallest possible number of actions. Additionally, the number of steps per episode T is limited by $T \leq (2N_c + 1)$. At the end of each episode, the additional reward is calculated as:

$$R = \begin{cases} \frac{1}{N_c}\sum_1^{N_c} r_i & \text{if } \min_{\forall m\in\boldsymbol{m}} m \geq m_{\mathrm{Th}} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

using the reward per channel

$$r_i = \frac{\|[m_{i\sim}, b_i] - [m_{\max\sim}, b_{\min}]\|}{\|[0, b_{\max}] - [m_{\max\sim}, b_{\min}]\|} \quad (4)$$

$$m_{\max\sim} = m_{\max} - m_{\mathrm{Th}}$$

$$m_{i\sim} = m_i - m_{\mathrm{Th}}$$

where $m_{\mathrm{Th}} = 1.7\,\mathrm{dB}$ is the minimum required margin and $m_{\max}$ the maximum margin, which is set heuristically.

Since the optimal parameters of the transmission system and the maximum aggregate bit rate are unknown during training, the maximum per channel bit rate $b_{\max} = 600\,\mathrm{Gb/s}$ is set to the highest possible throughput of the transponders used in the simulation. To

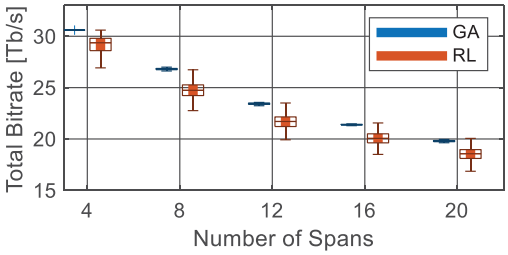

Fig. 2: Overall bitrate achieved by the agent in 1000 solutions compared to 53 solutions found with a GA for span counts $[4, 8, 12, 16, 20]$.

evaluate the performance of this approach the simulation setup is implemented based on the available advantage actor critc (A2C) algorithm[12] from the stable baseline library[13]. The A2C algorithm utilizes two neural networks to parameterize the state-value function and the stochastic policy of the agent. The agent is trained to maximize the overall bit rate for 51 channels of the transmission system as shown in Fig. 1, using N_s spans of $80\,\mathrm{km}$ LEAF. It is implemented using feed-forward neural networks (FNNs) with 64 and 32 neurons in the hidden layers, respectively. At the beginning of each episode the state of the environment is initialized with $200\,\mathrm{Gb/s}$ and $-3\,\mathrm{dBm}$ for each channel and the number of spans is uniformly distributed in the range of 4 to 23. The number of episodes N_E is set to $100{,}000$. Results are shown for a range of 1000 testing solutions. For each solution the agent starts at the initialized state as in training, but with a fixed number of spans N_s and tries to maximizes the bitrate. If one margin of the channels is below m_{Th}, the solution is considered as unfeasible. The trained agent is compared to solutions produced by optimizing the system parameters with a GA, which has been trained for 500 generations and a population size of 64.

Results and Discussion

As shown in Fig. 2 the overall achieved bitrate of the RL agent can compete and sometimes surpass the solutions produced by the GA, even though the variance of the overall bitrate is significantly higher. The influence of optimizing the hyperparameters of the FNNs and increasing N_E could be investigated to adress this issue.
It is noticeable, that the computation time differs significantly between these two approaches. The GA produces a single solution within 5-10 min, whereas a RL agent takes 1s on average, after 12h of initial training on our state-of-the-art PC.

Another important advantage of the RL is its generalization capability. An RL agent is able to produce solutions for scenarios/states for which it has not been trained explicity. This principle is shown in Fig. 3, where four different RL agents are compared, and each agent has been trained on a subset of span counts. The comparison of all trained agents shows that there is no significant difference or decrease in the predicted maximum possible throughput, but the percentage of invalid solutions varies with each agent. Furthermore, it can be seen that agents trained for less span counts (every 4th, every 8th) tend to perform better than the agent trained for each span, especially for low span counts. The agents choose riskier solutions, which lead to higher bitrates, but decreases the percentage of valid runs. Less span counts result in a smaller and therefore better trained state space of the agent. And the generalizability of RL then leads to the fact that good results can be achieved for other spans counts as well.

Conclusion

We investigated a RL algorithm to maximize the overall bitrate of a transmission system and compared it to a GA. The predicted optical performance of the algorithms is very similar, whereby the higher variance of the RL algorithm might be reduced by hyper parameter optimization. However, RL convinces by its ability to generalize, which means that almost identical results can be achieved with less training. Future work targets are to add further parameters to the state space, such as span length or fiber type, as well as the optimization of the neural networks.

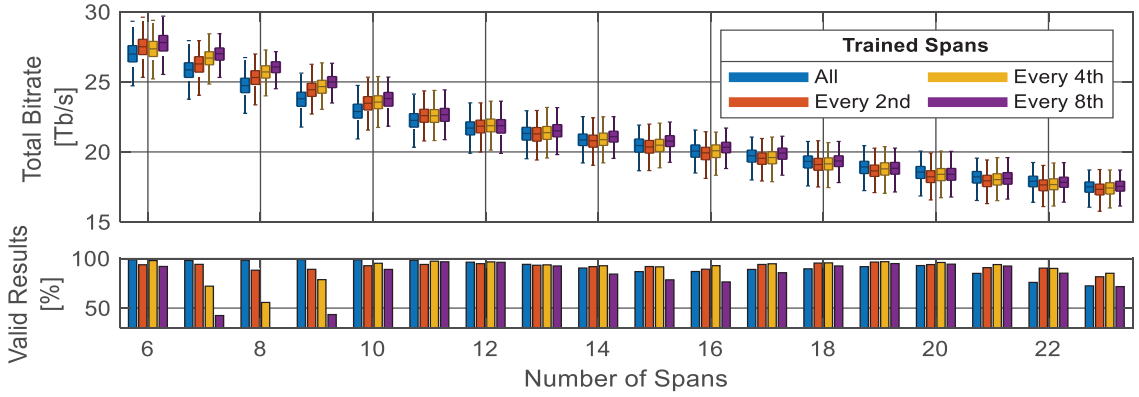

Fig. 3: Overall bitrate and percentage of valid results achieved by the agent in 1000 solutions. The agents were trained for each, every second, fourth and eighth span and tested for every span in the range of 6 and 23.

References

[1] L. Yan, *et al.*, “Link-Level Resource Allocation for Flexible-Grid Nonlinear Fiber-Optic Communication Systems”, *IEEE Photonics Technology Lett.*, vol. 27, no. 12, pp. 1250-1253, June, 2015,

[2] P. Poggiolini, “The GN model of non-linear propagation in uncompensated coherent optical systems”, *J. Light. Technol.*, vol. 30, no. 24, pp. 3857–79, 2012.

[3] A. Carena, *et. al.*, “EGN model of non-linear fiber propagation”, *Opt. Express*, vol. 22, no. 13, pp. 16 335–16 362, Jun. 2014.

[4] P. Poggiolini, *et al.*, “A Simple and Accurate Closed-Form EGN Model Formula”, *arXiv:1503.04132*, no. 12, pp. 1–5, 2015.

[5] D. J. Ives, S. J. Savory, “Transmitter optimized optical networks”, *Optical Fiber Communication Conference and Exposition and the National Fiber Optic Engineers Conference (OFC/NFOEC)*, Anaheim, CA, 2013, pp. 1-3.

[6] L. Yan, *et al.,* “Resource allocation for flexible-grid optical networks with nonlinear channel model [invited]”, *IEEE/OSA Journal of Optical Communications and Networking*, vol. 7, no. 11, pp. B101-B108, Nov. 2015

[7] Xiang Zhou, *et al.*, “Dynamic RMSA in elastic optical networks with an adaptive genetic algorithm”, *IEEE Global Communications Conference (GLOBECOM)*, Anaheim, CA, 2012 pp. 2912-2917.

[8] X. Chen, *et al.*, “Deep-RMSA: A Deep-Reinforcement-Learning Routing, Modulation and Spectrum Assignment Agent for Elastic Optical Networks”, *Optical Fiber Communications Conference and Exposition (OFC)*, San Diego, CA, 2018, pp. 1-3.

[9] M. Mitchell. An Introduction to Genetic Algorithms. Cambridge, MA: MIT, 1998.

[10] R. S. Sutton, A. G. Barto. Introduction to Reinforcement Learning (1st. ed.). MIT Press, Cambridge, MA, USA, 1998.

[11] H. Ali, *et al., “*A view on deep reinforcement learning in system optimization,” *arXiv:1908.01275,* 2019.

[12] V. Mnih, *et al.,* “Asynchronous methods for deep reinforcement learning”, *Proceedings of the 33nd International Conference on Machine Learning (ICML*), New York City, NY, USA, 2016, pages 1928–1937.

[13] A. Hill, *et al.*, Stable baselines. https://github.com/hill-a/stable-baselines,2018.