

Copyright Notice

© 2025 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

This material is presented to ensure timely dissemination of scholarly and technical work. Copyright and all rights therein are retained by authors or by other copyright holders. All persons copying this information are expected to adhere to the terms and constraints invoked by each author's copyright. In most cases, these works may not be reposted without the explicit permission of the copyright holder.

Institute of Communication Networks and Computer Engineering
University of Stuttgart
Pfaffenwaldring 47, D-70569 Stuttgart, Germany
Phone: ++49-711-685-68026, Fax: ++49-711-685-67983
Email: mail@ikr.uni-stuttgart.de, <http://www.ikr.uni-stuttgart.de>

The Impact of Reinforcement Learning Formulations on Solving the Routing and Spectrum Assignment Problem

Filippos Christou 

*Institute of Communication Networks
and Computer Engineering (IKR)
University of Stuttgart
Stuttgart, Germany
filippos.christou@ikr.uni-stuttgart.de*

Nicolas Hornek 

*Institute of Communication Networks
and Computer Engineering (IKR)
University of Stuttgart
Stuttgart, Germany
nicolas.hornek@ikr.uni-stuttgart.de*

Andreas Kirstädter

*Institute of Communication Networks
and Computer Engineering (IKR)
University of Stuttgart
Stuttgart, Germany
andreas.kirstaedter@ikr.uni-stuttgart.de*

Abstract—Every day, an increasing number of services rely on the efficient operation of communication networks. Although these networks have grown overly complex, their principal purpose will always remain to enable data exchange. IP-Optical networks play a pivotal role in worldwide data transmission, for which the routing and spectrum assignment/allocation (RSA) problem must be solved. While useful, traditional optimization methods and heuristics fall short under certain conditions, urging to consider reinforcement learning (RL) due to its adaptability and efficiency in handling complex decision-making problems. This paper extensively analyzes various partially observable Markov decision process (POMDP) formulations and their impact on solving the RSA problem while always using proximal policy optimization (PPO) as the underlying RL algorithm. We evaluate these formulations across different network topologies and demand patterns, benchmarking the performance of RL against baselines such as k-shortest path first-fit (KSPFF), integer linear programming (ILP), and the random policy. Our findings confirm critical dependence on the formulation, which, when done properly, RL can match or outperform the baselines. Notably, POMDPs with fewer possible actions and more concise observations improve the RL agent’s performance. The best POMDP formulations also exhibit consistent performance across multiple topologies, and the differently defined reward signals do not affect the overall performance. The study presents such and similar findings using a carefully planned workflow with close attention to statistical significance, appropriate baselines, and comprehensive visualizations.

Index Terms—reinforcement learning, IP-optical networks, routing and spectrum assignment

I. INTRODUCTION

The world has largely embraced and integrated the Internet indispensably into everyday life. What was once a niche and specialized service for a few has now become a global platform that breaks down barriers, connects people, and is often discussed as a human right. The Internet has become essential to today’s infrastructure, and numerous governments, businesses, and individuals depend on it. Although the Internet revolution is still in progress as big digitization projects and

regulations are ongoing, broad connectivity is already widely spread. However, the next big thing approaching us, i.e., artificial intelligence (AI), has not yet gained the same level of confidence. Even if the rise of AI has resulted from a yearslong effort from numerous institutions, big companies have led the AI boom over the previous years. The broad availability of large AI models has disrupted the status quo and sparked both excitement and concern. Besides the societal impact, one of the main concerns is the struggle to understand, trust, and handle AI models. The same concerns must be addressed across every discipline practicing AI/machine learning (ML) algorithms.

In IP-optical networks, there is a constant endeavor to maximize traffic deployment while keeping resource usage low. Considering both IP routing and optical switching gives rise to the routing and spectrum assignment/allocation (RSA) problem, which is notoriously known for being \mathcal{NP} -hard. Many previous works have attempted to solve the RSA problem using either traditional optimization methods (e.g., integer linear programming (ILP)), metaheuristics (e.g., genetic algorithms), heuristics (e.g., k-shortest path first-fit (KSPFF)), or ML algorithms.

By leveraging ML, the research community wishes to achieve the efficiency of strict optimization techniques while maintaining the speed of heuristics. Many researchers prefer reinforcement learning (RL) due to the absence of labeled datasets, the high dynamicity, and the sheer complexity of the given problem. Modern research in the area is characterized by aggressive cross-domain efforts, where new potent RL algorithms and techniques are adapted to solve the RSA or similar problems.

In this work, we take a step back, revisit basic concepts, and diligently investigate the process of using RL techniques to solve the RSA problem. Namely, we document the influence of different RL formulations, i.e., partially observable Markov decision processes (POMDPs), on solving RSA. We analyze the behavior across a series of topologies and demand patterns. Moreover, we compare the RL results with the commonly used KSPFF, a randomized allocation, and the definite optimum obtained from an ILP. Along these lines, instead of specializing in a single RL approach, we focus on

This work has been performed in the framework of the CELTIC-NEXT EUREKA project AI-NET-ANTILLAS (Project ID C2019/3-3), and it is partly funded by the German BMBF (Project ID 16KIS1312).

the fundamentals and conduct extensive research to extract valuable lessons that will help drive future efforts in the field.

The following section introduces the basics of IP-optical networks and RL. Section III presents a curated set of the most relevant literature and how this work differs. Section IV enumerates the involved POMDP formulations. Section V describes the simulation setup before we proceed with the results in Section VI. We conclude in Section VII.

II. BACKGROUND

A. IP-optical networks

IP-optical networks are multilayer networks that combine the IP layer, the optical layer, and their equipment. During a traffic request, a decision must be made at the IP, e.g., choose the routing path, and at the optics level, e.g., choose the frequency of the modulated signal in the fiber. Before elastic optical networks (EONs) [1], when the spectrum was divided into a fixed grid, and only certain wavelengths were available, operators needed to solve the routing and wavelength assignment (RWA) problem [2]. With EONs and the introduction of the flex grid spectrum, which enabled the selection of arbitrary frequencies and bandwidths per connection, the RSA problem emerged. Since then, several extensions of RSA have appeared in the literature. RSA can extend to problems such as Grooming, Routing, Modulation, Band, Spectrum, and Core assignment (GRMBSCA) [3] [4] [5]. Moreover, these problems can often be tackled in conjunction with other considerations like energy consumption [6] or latency reduction [7]. In this groundwork, we focus on the pure RSA problem, which remains the core of all these extensions.

B. Reinforcement learning

Unlike supervised learning, where an approximator is trained on labeled samples, RL involves learning through interaction with the environment in a closed feedback loop through rewards. Within this loop, the agent learns a function called policy to select an action given an observation. The learning process is characterized by a trial-and-error approach, carefully balancing exploration against exploitation. Exploration refers to taking random actions to witness new state trajectories. Exploitation means taking greedy actions intended to maximize the expected cumulative future rewards. The agent interacts with the environment by taking actions, observing resulting states, and receiving rewards. The system must be a (PO)MDP consisting of state S , observation O , action A , reward R , and transition dynamics T . Proximal policy optimization (PPO) [8] is a state-of-the-art algorithm that holds great promise in solving RL problems, which we also adopt in this work. It uses a clipped policy objective term to increase training stability. The overall objective function consists of a value estimation error to be minimized, the clipped policy objective to be maximized, and an entropy bonus to improve exploration. A common technique to improve an RL agent's performance is invalid action masking. Instead of the agent learning which actions are invalid in a given state, they are filtered out and removed from the agent's choices [9].

III. RELATED WORK

Throughout the years, multiple RL approaches, including different POMDP formulations, have been used to solve the RSA problem and improve the efficiency of IP-optical networks. The authors of [10] use an observation consisting of spectrum statistics along paths and let the agent choose among a subset of available actions. [11] proposes adding the $n \in \mathbb{N}^+$ possible next observations to the current one. Each observation contains path-level features like the remaining capacity. [12] uses a technique called semi-flexible spectrum assignment to reduce the dimensionality of the observation. [13] proposes a scheme for action masking that allows demands to be placed only directly next to already occupied spectrum slots and uses fragmentation-based reward metrics across links and slots. In [14], the latency is part of the observation and reward. [15] presents general techniques for applying RL to RSA, such as invalid action masking and the challenges long episodes pose, called the credit assignment problem. In [16], a convolution-based positional encoding for the spectrum slot matrix is proposed. Recently, graph neural networks (GNNs) have been introduced as an algorithmic advancement. [17] uses GNNs, one individual agent per node, and a fragmentation-based reward inspired by [18]. [19] uses GNNs to construct a topology agnostic agent.

The current work distinguishes itself by proposing a novel treatment of the problem by simultaneously evaluating the RL agent over an extended amount of simulation parameters and insightful baselines. Namely, we define a series of diverse POMDP formulations and compare them across different topologies and demand patterns. We use baselines like random policy and ILP, not seen for benchmarking an RSA RL agent before. This paper shifts focus from exclusively optimizing single problem instances to studying hundreds of them in a necessary attempt to revisit basic concepts.

IV. POMDP FORMULATIONS

First, we introduce the POMDPs involved in this study. Let $G(V, E, F)$ be a directed EON topology graph with V nodes and E edges. Each edge is a fiber consisting of F frequency/spectrum slots with e_f indicating an available $e_f = 1$ or an occupied $e_f = 0$ spectrum slot $f \in F$ of edge $e \in E$. The vector e_F^e expresses the availability of all spectrum slots F in edge e and can be extended to a matrix $\mathbf{E}_F^{|E| \times |F|}$ that stores the availability over all edges E and all spectrum slots F . The set U contains all source and destination pairs $U = \{(v_1, v_2) \in V \times V : v_1 \neq v_2\}$ as integer tuples. The same information can be represented as a tuple of one-hot vectors $\vec{u} = \{v_1^-, v_2^-\}$, where instead of an integer $z \in V$, we have a vector $\vec{v} \in \{0, 1\}^{|V|}$ with $v_i = 0 \forall i \in V \setminus \{z\}$ and $v_z = 1$. Let $k, j, b \in \mathbb{N}^+$ be contained in the sets K, J, B respectively, with each set containing numbers starting from 1 up to the maximum $\max\{K\} = |K|$, $\max\{J\} = |J|$, and $\max\{B\} = |B|$, where $|\cdot|$ denotes the cardinality of the set. We precompute the first $|K|$ shortest paths $p_{u,k} \in P_u$. The set P_u , with $|P_u| = |K|$ contains all $|K|$ shortest paths of node pair $u \in U$ and P is the superset containing all $|K|$ shortest paths across all node pairs U , with $|P| = |U| \cdot |K|$. Similarly to e_f, e_F^e , and \mathbf{E}_F , we define p_f, \vec{p}_F , and $\mathbf{P}_F^{|P| \times |F|}$.

to denote the spectrum availability across paths $p \in P$ instead of edges. The path spectrum availability is constructed such that the spectrum continuity constraint is respected. The network receives demands $d_u \in \mathbb{N}^+$, indicating the number of spectrum slots needed between the nodes of $u \in U$. $p_j \in \mathbb{N}$ denotes the index of the j -th available spectrum slot on a path p , such that the demand d_u can be deployed starting at this slot, with $p_j = 0$ if it does not exist. The vector \vec{p}_j contains the first $|J|$ available spectrum slots, and the matrix $\mathbf{P}_J^{|P| \times |J|}$ does the same across all the paths. Moreover, p_b denotes the b -th available spectrum block of path p , such that a demand d_u can fit. Fig. 1 shows how available spectrum slots are combined into a single available spectrum block. A spectrum block is defined using the index of the first available slot b_f and the overall block length b_l , i.e., $p_b = \{b_f, b_l\}$, with $b_f = b_l = 0$ if it does not exist. In that regard, we similarly define \vec{p}_b and $\mathbf{P}_B^{|P| \times |B|}$. Aside from \mathbf{P}_F , \mathbf{P}_J , and \mathbf{P}_B , we define respectively $\mathbf{P}_F^{|K| \times |F|}$, $\mathbf{P}_J^{|K| \times |J|}$, and $\mathbf{P}_B^{|K| \times |B|}$ as submatrices that only contain the available slots/blocks of paths $p \in P_u$. Based on the previous definitions, we can define some statistical measures, e.g., \bar{p}_{b_l} is the average block length, and $\text{sum}\{\vec{p}_F\}$ is the number of all available spectrum slots on path p .

All POMDPs have the same state, S , which is hidden from the agent, and contains the complete knowledge of the system, i.e., the \mathbf{E}_F , the current time step t , and a generator for the current and future demands d_u . What remains to be defined for every POMDP is the observation O , the actions A , the reward R , and an extra action generating and masking function $g: \{U, O, A\} \mapsto A': A' \subseteq A$. If g is not explicitly introduced, an action masking that simply outputs valid actions is applied, allowing only available paths and slots for the node pair $u \in U$. An action $a \in A$ consists of a selected path containing the involved edges $a_p = \{e: e \text{ belongs to path } p, e \in E\}$ and a spectrum slot a_f from \mathbf{P}_F^u , \mathbf{P}_J^u , or \mathbf{P}_B^u . Let O^t notate the agent's observation at time step t . The transition dynamics T are considered partially known to the agent in that allocating specific spectrum slots will yield states with these slots being occupied. However, the future arrival, departure, and value d_u of the demands are unknown to the agent. All the POMDPs solve the same RSA problem (game) without grooming or electrical-optical-electrical (E-O-E) signal regeneration. The network progressively receives demands, and the game terminates when the agent blocks. These game rules have been preferred over others in the literature where the game continues after blocking. After continuous blocking the spectrum is broadly occupied and the agent's subsequent masked actions are usually trivial, leading to reduced learning (i.e., inefficient sampling).

- $POMDP_1$ $\color{magenta}{\blacklozenge}$: $O = \{\mathbf{E}_F, u, d_u, t\}$, $A = \mathbf{P}_F^u$
- $POMDP_2$ $\color{red}{\blacktriangle}$: $O = \{\mathbf{P}_B^u, u, d_u, t\}$, $A = \mathbf{P}_F^u$

The function g narrows down the actions A to the valid ones that have spectrum slots contained on the B spectrum blocks.

- $POMDP_3$ $\color{green}{\blackstar}$: $O = \{\mathbf{P}_F^u, u, d_u, t\}$, $A = \mathbf{P}_F^u$
- $POMDP_4$ $\color{blue}{\blacksquare}$: $O = \{\mathbf{P}_J^u, u, d_u, t\}$, $A = \mathbf{P}_J^u$
- $POMDP_5$ $\color{orange}{\times}$: $O = \{\mathbf{E}_F, u, d_u, t\}$, $A = \mathbf{P}_J^u$
- $POMDP_6$ $\color{red}{\bullet}$: $O = \{\mathbf{P}_F^u, u, d_u, t\}$, $A = \mathbf{P}_J^u$

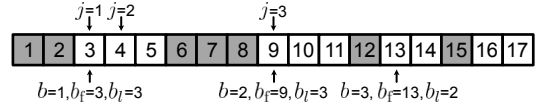


Figure 1. The difference between $|B|=3$ available spectrum blocks and $|J|=3$ available spectrum slots when $d_u=2$. White slots are available. Here, $\bar{p}_{b_l} = (3+3+2)/3 = 8/3$ and $\text{sum}\{\vec{p}_F\} = 3+3+2+2 = 10$

- $POMDP_7$ $\color{red}{\blacktriangle}$: $O = \{\mathbf{P}_B^u, \vec{u}, d_u, t, M_u\}$, $A = \mathbf{P}_B^u$

With $M_u = \{\vec{p}_{b_l}, \text{sum}\{\vec{p}_F\} \forall p \in P_u\}$. This POMDP is similar to [10]. The function g chooses the first spectrum slot b_f from each spectrum block b .

- $POMDP_8$ $\color{blue}{\blackstar}$: $O = \{\mathbf{P}_F^u, u, d_u, t, M_u\}$, $A = \mathbf{P}_F^u$
- $POMDP_9$ $\color{green}{\blacksquare}$: $O = \{\vec{O}^{t+1}, u, d_u, t\}$, $A = \mathbf{P}_J^u$

This is the unique POMDP that uses the known dynamics T , to calculate a vector of possible next observations \vec{O}^{t+1} produced by taking actions $A = \mathbf{P}_J^u$. Each possible future observation has the form $O^{t+1} = M_u$. This POMDP is similar to [11].

- $POMDP_{10}$ $\color{orange}{\times}$: $O = \{\vec{O}^{t-n}, \dots, O^{t-i}, \dots, O^t\}$, $A = \mathbf{P}_F^u$

This POMDP uses the current and the n previous observations based on $POMDP_1$, i.e. $O^{t-i} = \{\mathbf{E}_F^{t-i}, u^{t-i}, d_u^{t-i}, t-i\}$, with the superscript $t-i$ indicating the value at this time step.

- $POMDP_{11}$ $\color{cyan}{\bullet}$: $O = \{\vec{O}^{t-n}, \dots, O^{t-i}, \dots, O^t\}$, $A = \mathbf{P}_J^u$

Based on $POMDP_4$, i.e. $O^{t-i} = \{\mathbf{P}_J^{u^{t-i}}, u^{t-i}, d_u^{t-i}, t-i\}$

- $POMDP_{12}$ $\color{red}{\blacktriangle}$: $O = \{\vec{O}^{t-n}, \dots, O^{t-i}, \dots, O^t\}$, $A = \mathbf{P}_B^u$

The observations, and function g are based on $POMDP_7$, i.e. $O^{t-i} = \{\mathbf{P}_B^{u^{t-i}}, \vec{u}^{t-i}, d_u^{t-i}, t-i, M_u^{t-1}\}$

As the reward R is orthogonal to the rest of the formulation, we present them independently. We define the reward signal to be 0 if the demand can not be allocated and r otherwise, with the following options:

- $r_{\text{const}} = 1$
- $r_{\text{dem}} = d_u$ [19]
- $r_{\text{hops}} = \frac{1}{|a_p|}$ [14]
- $r_{\text{quang}} = 1 + e^{-H_{\text{frag}}}$ [17],
- $r_{\text{shimoda}} = r_{\text{linkfrag}} + r_{\text{slotfrag}}$ [13].

where H_{frag} is the Shannon-Entropy-Fragmentation from [18] and r_{linkfrag} , r_{slotfrag} are link and slot fragmentation metrics.

V. SIMULATIONS AND BASELINES

This section describes the simulation setup and the baselines used to benchmark against the RL agent. We applied the combinations of the POMDPs and rewards from Section IV across the topologies shown in Fig. 2 from [20]. The simulations run for $|K|=5$, $|B|=3$, $|J|=3$, and $|F|=50$. We use three different demand patterns designed to block the network at its steady state. Fig. 3 shows one *demand list* produced by each of them. The x -axis divides the simulation time into 25 windows, and the y -axis counts the number of demands included in that time window. On the left, *constanterlang* has a fixed ratio of arrival rate and service rate. In the middle, *oscillating* has a periodic arrival rate, producing a repeating profile. On the right, the *mouselephant* has a disproportional amount of higher value demands. The simulations' layout is shown in Fig. 4. A hyperparameter optimization was done for the reward signal r_{const} to

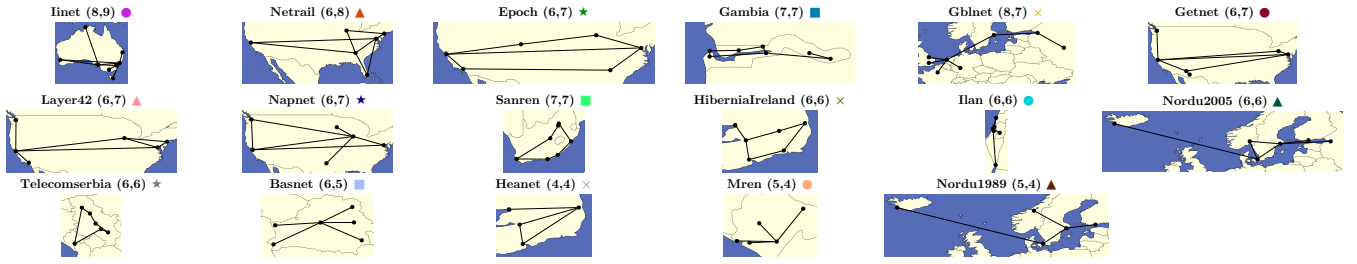


Figure 2. Topologies with $(|V|, |E|)$

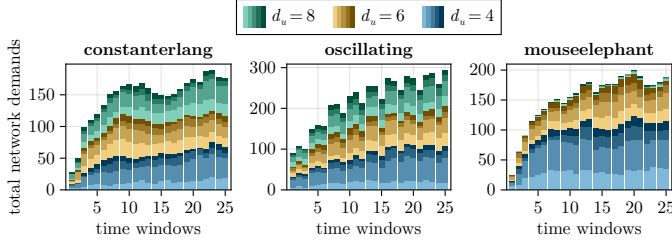


Figure 3. Demand patterns. The gradient colors for each demand value start from light, indicating a low hop count, to darker, with a higher hop count for deploying the given demand in the shortest hop path.

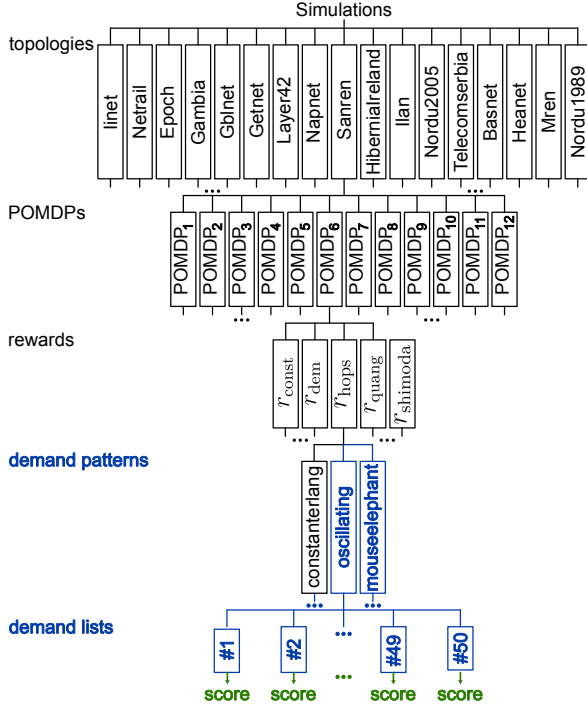


Figure 4. Simulations tree layout

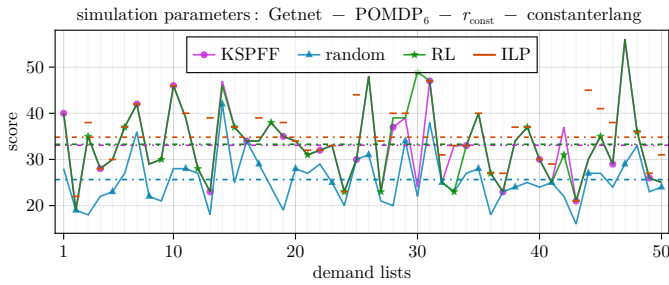


Figure 5. RL and baselines scores across 50 simulation leaves with a common parent from Fig. 4. The horizontal dashed lines are the mean value.

find a performant configuration. These PPO hyperparameters are used for all simulations with different rewards.

The agent is trained using only the *constanterlang* demand pattern. Then, the agent's policy is evaluated across 50 pre-generated unseen demand lists for each demand pattern and topology. To measure the *score*, we count the number of demands served before blocking. Fig. 5 shows the RL and baseline scores for a single problem instance across the evaluation dataset. The KSPFF score is calculated by picking the shortest $k \in K$ available path in P_u and the first available spectrum slot $j \in J$. The random score is calculated by taking random valid actions using g for the specific POMDP. We notice the variance in score over the different demand lists. These fluctuations explain why keeping the demand lists fixed across the baselines is critical. The ILP score is the optimum score that can be achieved. Although an unfair comparison, since the ILP model knows the sequence of all future demands in advance, it provides excellent insights as the theoretical maximum. Due to space reasons, the ILP model is not presented here. As the complexity of the problem is vast, the ILP could not provide an optimum solution for some cases using a timeout of 2 days. For similar reasons, we focused on small topologies and spectrum bands, such that the RL hyperparameter optimization and training finish within a reasonable time for all simulations. Scaling the solution to bigger spectrum bands is not studied here but could be done using a multi-agent approach, where the same learned policy is applied across different spectrum slot windows of length $|F| = 50$.

VI. RESULTS

This section presents and discusses our results after running and evaluating all the simulations above. Different topologies and demand patterns have inherently different blocking rates. Therefore, it is more appropriate to look at the results in a relative manner. The relative win is more interesting, as this study does not focus on finding the best RL approach but on investigating how the different POMDPs behave under diverse circumstances.

Fig. 6 demonstrates this using the relative win (RW) of the RL score over the KSPFF as a metric:

$$RW_{KSPFF}^{RL} = \frac{(\text{score}_{RL} - \text{score}_{KSPFF})}{\text{score}_{KSPFF}}$$

It plots the average value of the relative win in a heatmap by focusing on the *constanterlang* demand pattern and the r_{const} reward. We notice that the POMDPs performing consistently

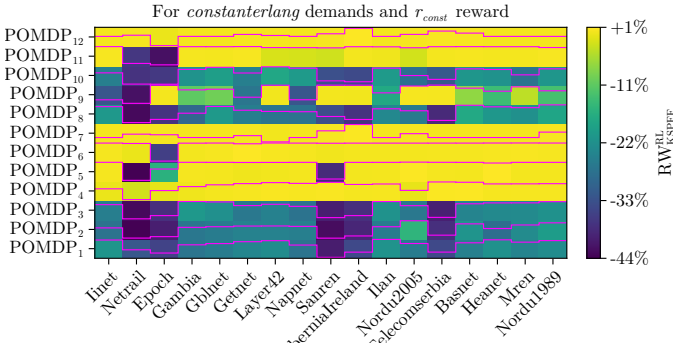


Figure 6. Impact of the topologies on the POMDPs. The magenta stairs-plot lines are drawn per POMDP and evenly scale the heatmap row values from minimum on the bottom to maximum on the top, such that slight differences along a row are easier to spot.

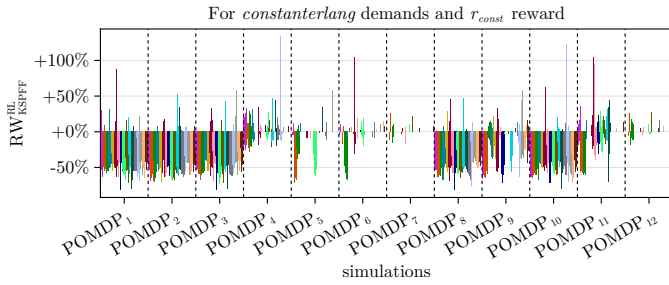


Figure 7. Relative win of Fig. 6 for all data points. Topologies are colored according to the symbol of each topology title in Fig. 2.

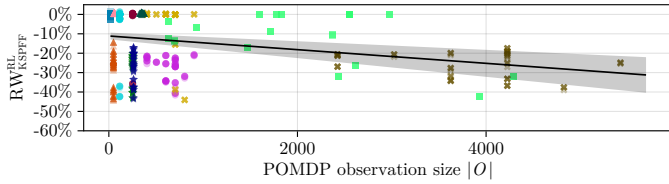


Figure 8. Linear regression of RW_{KSPFF}^{RL} over the POMDP observation size. The markers indicate the topology used based on the topology title markers in Fig. 2.

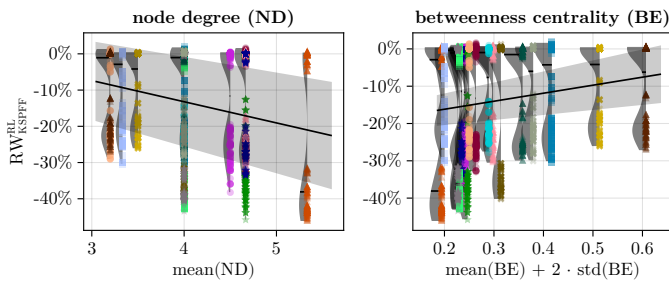


Figure 9. Linear regression of RW_{KSPFF}^{RL} over the topology properties. The markers indicate the POMDP used based on the markers in the list of Section IV.

across all topologies are also the best ones overall. POMDP₉ is an interesting case that selectively performs better in topologies that are pitfalls for others. An important observation is that the POMDPs with fewer available actions are doing better. This can be unintuitive since more available actions give the agent more freedom. Yet, this increased search space often leads to inefficient sampling, practically locking the agent to local minima. Also, concatenating the current observation with past

ones does not seem to yield better results. This may be because the current observation already identifies the hidden state well enough, such that additional past observations overwhelm the agent. Moreover, ring topologies like Telecomserbia and Sanren have generally been harder to solve well.

Fig. 7 plots the same data but spread across the demand lists without taking the mean. This figure is a testament to the importance of statistical significance during experiments. Although there are cases where RL exceeds KSPFF by more than 100%, on average, this relative win does not go higher than 1%, as the color bar in Fig. 6 shows.

In Fig. 8, we aim to find a numerical relationship between the POMDPs and RW_{KSPFF}^{RL} . Namely, we show that in our experiments, the bigger the observation size, the worse the RL agent performs. This trend is shown with a linear regression and a confidence interval of 99%. The negative correlation might be unexpected as a more informed RL agent with a larger observation size should comprehend the problem better. However, the computational nature of RL in a world without infinite computational resources calls for a reduced dimensionality of the problem. Likewise, Fig. 9 shows a relationship between the topology properties and RW_{KSPFF}^{RL} . On the left, the average node degree of the topology is used as a predictor. A higher node degree corresponds to a more complex topology, which the RL algorithm struggles with. On the right of Fig. 9, the predictor is the mean betweenness centrality (BC) of the graphs' edges plus two times its standard deviation. The predictor is engineered to indicate, on average, how easy it is to traverse the network (mean BC) while keeping the topology diverse (standard deviation of BC), e.g., in contrast with ring topologies. Indeed, this predictor successfully produces a clearly positive correlation with RW_{KSPFF}^{RL} .

Fig. 10 shows the deviation of the random, KSPFF, and RL scores relative to the ILP scores, using Gaussian kernel estimators to draw their distributions. We observe that the random scores can sometimes approach the optimum solution near 0%. The random policy is equal to the untrained RL agent. So, this graph also shows the effect of the training on the RL agent. For this reason, researchers developing RL algorithms should generally include the random policy as a baseline. Some POMDPs, like POMDP₇, can get well ahead just by using a random policy with a sophisticated action generating and masking function g . Lastly, this figure also tells us that the KSPFF score is often equal to the optimal one, explaining why KSPFF can be so hard to outperform significantly using RL.

Although the RL agent is only trained on *constanterlang* demands, it performs similarly on different demand patterns. To

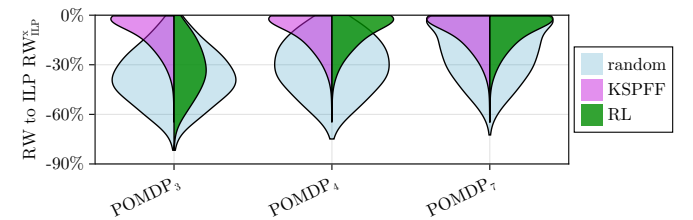


Figure 10. Scores of random, KSPFF, and RL related to ILP across all topologies, rewards, and demand patterns. Similar for all POMDPs.

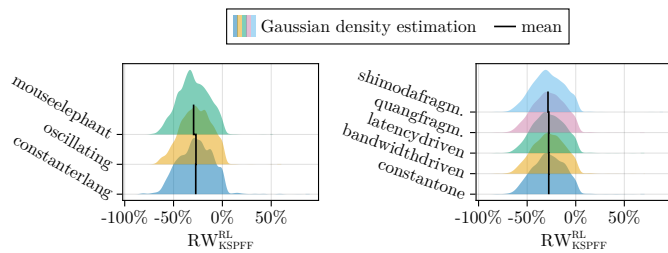


Figure 11. Different demand patterns and rewards r barely make any difference. Data generated using POMDP₃. Similar for all POMDPs.

confirm that, we calculate the RW_{KSPFF}^{RL} for all demand patterns and compare them with one another. In particular, the RL agent does barely worse only when exposed to *mouseelephant* traffic (on average -8%). The plot on the left of Fig. 11 demonstrates this for a single POMDP. The same calculations are applied for different rewards on the right of Fig. 11. Using different rewards produces no significant score variation and indicates a valid hyperparameter optimization where the hyperparameters for r_{const} successfully transfer to other rewards. Fig. 10 and Fig. 11 look similar for all POMDPs.

VII. CONCLUSIONS

We presented a rigorous workflow for studying the application of RL to the RSA problem. We used an extensive and fixed evaluation dataset across different simulation parameters and baselines to guarantee statistical significance. We underline the importance of calculating the theoretical optimum (ILP) value when possible to be aware of the improvement potential of the RL approach. Moreover, the random policy baseline should be documented to distinguish the RL agent’s performance from the benefits of action masking.

Building on these foundations, we evaluated several POMDP formulations across various topologies and demand patterns. Solving RSA using RL quickly becomes a computational problem, leading to the question of which POMDP formulation best conveys valuable information to the RL agent. In practice, smaller and more concise observations, as well as fewer available actions, proved to be more effective. Highly performant POMDPs consistently performed well across all the topologies. Using possible next observations in the POMDP representation emerged as a promising technique to tackle common pitfalls. However, concatenating the current observation with past ones did not yield better results. Notably, the agent’s performance remained consistent across the defined reward signals and proved robust against the demand patterns used. We found that topologies with lower node degrees are easier to solve, with ring topologies being an exception. Lastly, edge betweenness centrality correlates with higher performance. Future work in the field is crucial as it will unlock a deeper understanding of applying RL to RSA. Such could include variations of the RSA game rules, a wider set of POMDPs, alternative RL algorithms, and scaling to bigger problems.

REFERENCES

- [1] B. C. Chatterjee, N. Sarma, and E. Oki, “Routing and spectrum allocation in elastic optical networks: A tutorial,” *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1776–1800, 2015.
- [2] A. Ozdaglar and D. Bertsekas, “Routing and wavelength assignment in optical networks,” *IEEE/ACM Transactions on Networking*, vol. 11, no. 2, pp. 259–272, 2003.
- [3] L. C. Resendo, “Optimal approach for electronic grooming, routing and spectrum allocation in elastic optical networks,” in *2015 SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference (IMOC)*, 2015, pp. 1–5.
- [4] Q. Yao, H. Yang, B. Bao, J. Zhang, H. Wang, D. Ge, S. Liu, D. Wang, Y. Li, D. Zhang, and H. Li, “Snr re-verification-based routing, band, modulation, and spectrum assignment in hybrid c-c+l optical networks,” *Journal of Lightwave Technology*, vol. 40, no. 11, pp. 3456–3469, 2022.
- [5] Í. B. Brasileiro, L. R. Costa, and A. C. Drummond, “A survey on crosstalk and routing, modulation selection, core and spectrum allocation in elastic optical networks,” *ArXiv*, vol. abs/1907.08538, 2019.
- [6] Y. Tan, R. Gu, and Y. Ji, “Energy-efficient routing, modulation and spectrum allocation in elastic optical networks,” *Optical Fiber Technology*, vol. 36, pp. 297–305, 2017.
- [7] C. Hernández-Chulde, R. Casellas, R. Martínez, R. Vilalta, and R. M. noz, “Experimental evaluation of a latency-aware routing and spectrum assignment mechanism based on deep reinforcement learning,” *J. Opt. Commun. Netw.*, vol. 15, no. 11, pp. 925–937, Nov 2023.
- [8] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” 2017. [Online]. Available: <https://arxiv.org/abs/1707.06347>
- [9] M. Doherty, Y. Zhang, and A. Beghelli, “Masked Deep Reinforcement Learning for Virtual Network Embedding on Elastic Optical Networks.”
- [10] X. Chen, B. Li, R. Proietti, H. Lu, Z. Zhu, and S. J. B. Yoo, “Deepprmsa: 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.
- [11] J. Suarez-Varela, A. Mestres, J. Yu, L. Kuang, H. Feng, P. Barlet-Ros, and A. Cabellos-Aparicio, “Feature Engineering for Deep Reinforcement Learning Based Routing,” in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, May 2019, pp. 1–6.
- [12] R. Shiraki, Y. Mori, H. Hasegawa, and K.-i. Sato, “Dynamically Controlled Flexible-Grid Networks Based on Semi-Flexible Spectrum Assignment and Network-State-Value Evaluation,” in *Optical Fiber Communication Conference (OFC) 2020*. San Diego, California: Optica Publishing Group, 2020, p. M1B.4.
- [13] M. Shimoda and T. Tanaka, “Deep Reinforcement Learning-based Spectrum Assignment with Multi-metric Reward Function and Assignable Boundary Slot Mask,” in *26th Optoelectronics and Communications Conference*. Hong Kong: Optica Publishing Group, 2021, p. M4B.3.
- [14] C. Hernandez-Chulde, R. Casellas, R. Martinez, R. Vilalta, and R. Munoz, “Assessment of a Latency-aware Routing and Spectrum Assignment Mechanism Based on Deep Reinforcement Learning,” in *2021 European Conference on Optical Communication (ECOC)*. Bordeaux, France: IEEE, Sep. 2021, pp. 1–4.
- [15] J. W. Nevin, S. Nallaperuma, N. A. Shevchenko, Z. Shabka, G. Zervas, and S. J. Savory, “Techniques for applying reinforcement learning to routing and wavelength assignment problems in optical fiber communication networks,” *Journal of Optical Communications and Networking*, vol. 14, no. 9, pp. 733–748, Sep. 2022.
- [16] T. Tanaka and M. Shimoda, “Pre- and post-processing techniques for reinforcement-learning-based routing and spectrum assignment in elastic optical networks,” *Journal of Optical Communications and Networking*, vol. 15, no. 12, p. 1019, Dec. 2023.
- [17] H. T. Quang, O. Houidi, J. Errea-Moreno, D. Verchere, and D. Zeghlache, “MAGC-RSA: Multi-Agent Graph Convolutional Reinforcement Learning for Distributed Routing and Spectrum Assignment in Elastic Optical Networks,” in *2022 European Conference on Optical Communication (ECOC)*, Sep. 2022, pp. 1–4.
- [18] 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, pp. A66–A72, Jan. 2015.
- [19] P. Almasan, J. Suárez-Varela, K. Rusek, P. Barlet-Ros, and A. Cabellos-Aparicio, “Deep reinforcement learning meets graph neural networks: Exploring a routing optimization use case,” *Computer Communications*, vol. 196, p. 184–194, Dec. 2022.
- [20] S. Knight, H. Nguyen, N. J. G. Falkner, R. A. Bowden, and M. Roughan, “The internet topology zoo,” *IEEE Journal on Selected Areas in Communications*, vol. 29, pp. 1765–1775, 2011.