Techniques for applying reinforcement learning to routing and wavelength assignment problems in optical fiber communication networks

JOSH W. NEVIN[1,*], SAM NALLAPERUMA[1], NIKITA A. SHEVCHENKO[1], ZACHARAYA SHABKA[2], GEORGIOS ZERVAS[2], AND SEB J. SAVORY[1]

[1] *Fibre Optic Communication Systems Laboratory (FOCSLab), Electrical Engineering Division, Department of Engineering, University of Cambridge, 9 JJ Thomson Avenue, Cambridge CB3 0FA, U.K.*
[2] *Optical Networks Group, Department of Electronic & Electrical Engineering, University College London (UCL), Torrington Place, London WC1E 7JE, U.K.*
[*] *Corresponding author: jn399@cam.ac.uk*

Compiled August 8, 2022

We propose a novel application of reinforcement learning (RL) with invalid action masking and a novel training methodology for routing and wavelength assignment (RWA) in fixed-grid optical networks and demonstrate the generalizability of the learned policy to a realistic traffic matrix unseen during training. Through the introduction of invalid action masking and a new training method, the applicability of RL to RWA in fixed-grid networks is extended from considering connection requests between nodes to servicing demands of a given bit rate, such that light paths can be used to service multiple demands subject to capacity constraints. We outline the additional challenges involved for this RWA problem, for which we found that standard RL had low performance compared to baseline heuristics, in comparison with the connection requests RWA problem considered in literature. Thus, we propose invalid action masking and a novel training method to improve the efficacy of the RL agent. With invalid action masking, domain knowledge is embedded in the RL model to constrain the action space of the RL agent to lightpaths that can support the current request, reducing the size of the action space and thus increasing the efficacy of the agent. In the proposed training method, the RL model is trained on a simplified version of the problem and evaluated on the target RWA problem, increasing the efficacy of the agent compared to training directly on the target problem. RL with invalid action masking and this training method outperforms standard RL and three state-of-the-art heuristics, namely $k-$shortest path first fit, first fit $k-$shortest path and $k-$shortest path most utilized, consistently across uniform and non-uniform traffic in terms of the number of accepted transmission requests for two real-world core topologies, such as NSFNET and COST–239. The RWA run time of the proposed RL model is comparable to that of these heuristic approaches, demonstrating the potential for real world applicability. Moreover, we show that the RL agent trained on uniform traffic is able to generalize well to a realistic non-uniform traffic distribution not seen during training, outperforming the heuristics for this traffic. Visualization of the learned RWA policy reveals an RWA strategy that differs significantly from the heuristic baselines in terms of the distribution of services across channels and the distribution across links.

http://dx.doi.org/10.1364/ao.XX.XXXXXX

1. INTRODUCTION

The routing and wavelength assignment (RWA) problem in optical networks consists of selecting an optimal path and channel combination to transmit data between two requested nodes. RWA is proven to be a $\mathcal{NP}-$hard problem [1], meaning no exact approach exists that guarantees an optimal polynomial-time solution. Thus, there is considerable interest in new methods that improve on the efficacy of existing heuristic solutions. In the static RWA problem, it is assumed that all the requests are known before we start servicing them. For this case, it is possible to use global techniques such as integer linear programming to obtain an optimal solution. However, in many cases network operators do not have knowledge of all requests ahead of time. In this work we consider the sequential RWA problem, in which the requests arrive sequentially with an unknown distribution.

In this case, heuristics, such as $k-$shortest path first-fit channel (kSP-FF) are typically used to service the requests, as no global solutions are available. As RWA is a sequential decision-making problem it can be cast as a Markov decision process (MDP), and thus solved using reinforcement learning (RL), i.e., a machine learning (ML) technique in which a model, known as the agent, learns a solution by interacting with its environment via a feedback loop. RL has been shown to perform remarkably well for a vast range of control problems, from playing Atari games to a superhuman standard [2] to highly complex control tasks, such as controlling the plasma in a simulated nuclear fusion reactor [3]. One of the main benefits of RL is flexibility – in principle the same well-designed RL agent can learn a solution for a number of similar problems. In this sense, we can refer to RL as a general purpose method. Frameworks to solve classes of combinatorial optimization [4] and multi-objective optimization problems [5] based on RL have been proposed. In the case of RWA, this could mean we can train the same RL agent on different topologies and traffic matrices and obtain a good solution. In contrast, hand-tuned heuristics are problem-specific and must be designed for the target problem, with limited flexibility to changing conditions. Additionally, as RL agents learn through experience, they can discover solutions that are relatively free from human bias. However, problem-specific heuristics are designed based on the assumptions of the designer, meaning that their performance can suffer if these assumptions are limited.

Multiple previous studies have explored the efficacy of RL for solving RWA [6, 7] and related problems, such as routing assignment in fixed-grid networks [8, 9], routing, modulation and spectrum assignment (RMSA) in elastic optical networks [10–12] and IP/optical cross-layer routing [13]. The proposed models have a range of different design choices, with variations on how the network state is represented, the action space of the RL agent, the reward that is given to the agent and the RL algorithm used for training. However, for the fixed-grid case, these previous works have considered a relatively simple problem in which the RL agent learns how to service a series of connection requests [7, 14]. Thus, the agent learns how to establish lightpaths between requested node pairs, and each channel can only support a single connection request. Similarly, for flex-grid elastic optical networks, the spectrum slots considered can only support a single request at a time [10]. In this work, we consider fixed-grid RWA in terms of non-expiring demands of a given bit rate, such that a lightpath can be used to service multiple requests between the same two nodes, as long as it has sufficient remaining capacity. Our choice to model this system is motivated by systems that are currently deployed today. Thus, the agent has to learn not only how to establish lightpaths, but also how to reuse them. As a result of this, the network is able to support a far greater number of requests in our work compared to in relevant works from the literature. This means that the episode sizes, meaning the number of RWA decisions that must be made, considered in this work are much longer than those previously considered, by approximately a factor of 10. This is because it is possible to service many more requests, chosen to have a typical bit rate of 100 Gbps, before blocking occurs in our more realistic implementation. In general, longer episode sizes make RL problems more difficult as it becomes harder for the agent to identify which decisions were the most important in determining the final state. This is known as the credit assignment problem [15, 16]. Additionally, in our implementation lightpath reuse is possible, which has not been considered in the literature. Thus, the agent is required to learn not only how to optimally establish new lightpaths but also how to reuse existing ones in an optimal way. Thus, we present a novel application of RL to RWA problems with fixed bit rate requests, which is a more difficult problem for the reasons outlined above.

In this work, we present a novel action space for RWA in optical networks, in which invalid action masking is utilized. We demonstrate that masking invalid actions increases the performance of the agent as compared to without masking. Additionally, we consider a novel training mode for RWA in optical networks, in which the agent is trained on a simplified version of the target RWA problem with shorter episodes. We found that the policy learned for this problem generalizes to the target problem, achieving performance better than baseline heuristics commonly considered in the RL-driven RWA literature and increasing the performance relative to training directly on the target problem. Furthermore, we demonstrate that the RL agent trained on uniform traffic outperforms the heuristics with statistical significance of 99.5% when performing RWA on an unseen non-uniform traffic distribution, while heuristics fail to achieve consistent results across different traffic matrices. This demonstrates that the proposed RL solution has an ability to generalize to traffic distributions unseen during training. Moreover, the proposed RL approach shows applicability to real-world systems in terms of its RWA run-time, which is of the same order of the heuristic approaches considered.

We summarize our novel contributions here.

- We present a novel application of RL to the full RWA problem of servicing demands of a given bit rate over lightpaths with a given maximum capacity in fixed-grid networks. Thus, lightpaths can be used to service multiple demands.
- We demonstrate a novel application of RL with invalid action masking to RWA in optical networks.
- We show a novel application of a reduced complexity training method for RL-driven RWA in optical networks, reducing the difficulty of credit assignment during the training stage. This allows the agent to learn a policy that generalizes to a realistic target case with a higher efficacy as compared to training directly on that case.
- We perform interpretation of the learned RWA policy via visualization of the distribution of services as the episode progresses and comparison with the baseline heuristics. To the best of our knowledge, this is the first time such an interpretation is presented.

The organization of the remainder of the paper is as follows. Section 2 presents the state-of-the-art of RWA in optical networks and highlights the limitations in existing approaches. Section 3 describes the optical core network physical layer model used, followed by a summary of key RL theory and our proposed RL model in Section 4. The simulation set up is outlined in Section 5. We present the results of the RWA simulations for uniform traffic for the NSFNET and COST-239 topologies in Section 6, followed by results showcasing the generalization of the trained agent to a realistic traffic matrix unseen during training in Section 7. An interpretability study of the learned RWA policies is presented in Section 8, followed by concluding remarks in Section 9.

2. RELATED WORK

A. Routing and Wavelength Assignment in Optical Networks

RWA algorithms in optical networks select both the path taken through the network and the wavelength channel that is used for transmission for a given request to transmit data between two nodes. The path consists of a series of links that connect a given pair of nodes. Often, this choice is formulated as choosing between the $k-$shortest paths that connect the requested nodes, rather than all possible routes as these may be numerous. In this work, for instance, we consider the $5-$shortest paths between each node pair. RWA also involves selecting the wavelength that is used to transmit the data along the chosen path. Once a path and a wavelength is chosen, a lightpath is established on that path along the chosen wavelength. In fixed-grid wavelength division multiplexing (WDM) networks, there is always a fixed number of channels to choose from. As we are considering wavelength-routed WDM optical networks, the principle of wavelength continuity must be obeyed. RWA is proven to be a $\mathcal{NP}-$hard problem [1], meaning no exact approach exists that guarantees an optimal solution in polynomial time. In the literature several problem formulations have been proposed to solve RWA in optical networks [17, 18].

Conventionally, RWA is performed using polynomial time algorithms based on standard heuristics, such as $k-$shortest path first-fit wavelength (kSP-FF) [19]. In the literature the potential of reinforcement learning (RL) for routing assignment problems in optical networks has been demonstrated [8, 10–12, 20, 21]. This has been driven by a recent increase in the performance of reinforcement learning approaches [22]. A range of RL models have been proposed, aiming to find an solution that is optimal with respect to a range of performance metrics, such as network throughput [10], delay [8, 20], survivability [11], jitter and traffic volume [21]. Some approaches have considered the problem of routing, modulation and spectrum assignment (RMSA) in flex-grid elastic networks, where the agent must select the path, modulation format and spectrum that is used to service a given request [10–12]. The spectrum in such networks is composed of combinations of spectrum slots and the agent must learn to select the spectrum slots in an optimal way. On the other hand, other studies have considered fixed-grid WDM networks, in which wavelength channels with a fixed spacing are chosen by the agent [6, 7].

In these previous works, the RWA problem in fixed-grid WDM networks has been modeled in terms of connection requests, rather than servicing demands of a given bit rate. Thus, one lightpath can support only a single request, as the requests are simply to connect node pairs. In deployed networks, however, demands can be represented as a request to transmit data at a given bit rate between two nodes, rather than simply to connect them. We model this situation here, as it is a more complete representation of the RWA problem. This substantially increases the number of demands that must be serviced, as a lightpath with a given capacity can service multiple demands, thus increasing the difficulty of the allocation problem. As a result, we observed that additional RL techniques, namely invalid action masking and a simplified training phase were required to achieve results better than problem specific heuristics for this RWA formulation.

B. Domain Knowledge-Informed Machine Learning Approaches in Optical Networks

Invalid action masking constitutes a form of domain knowledge-informed ML, as we are utilizing what we already know about the system in order to reduce the size of the search space of the RL algorithm. A range of domain knowledge-informed ML approaches have been proposed in the optical networks literature, such as physics-informed neural networks (NNs) for nonlinearity estimation [23] and solving the nonlinear Schrödinger equation [24], and physics-informed Gaussian processes for regression problems in optical networks [25]. Domain knowledge has also been used to inform the design of ML approaches, by using the structure of the nonlinear Schrödinger equation to design NN architectures for learned digital backpropagation, e.g., [26]. A domain knowledge-informed RL approach to routing in IP/optical cross-layer networks has also been presented, in which an RL agent was enhanced by an experience-driven mathematical model of the system [13]. In these methods, domain knowledge in the form of the equations governing the physics of the optical fiber communications channel was embedded within ML approaches to improve their data efficiency and computational complexity. Thus, by using invalid action masking we employ a similar technique by embedding our domain knowledge of which lightpaths are able to support the current request into the action space of the RL agent.

3. NETWORK PHYSICAL LAYER MODEL

We make the simplifying assumption of transmission at the Shannon rate and assign the point-to-point throughput between a given source and destination node per lightpath as the theoretical upper-bound taken at the optimum launch power. The physical layer is modeled as a regular incoherent nonlinear interference Gaussian noise (GN) model [27] to assign the point-to-point throughput between a given source and destination node per lightpath. We also make the assumption that transmitted optical pulses have a rectangular spectrum, with channel bandwidth equal to the symbol rate. This is an idealized case, corresponding to the maximum spectral efficiency for fully-loaded links. In addition, we assume that the spectrum of nonlinear interference distortions is distributed across the modulated signal bandwidth as white Gaussian noise. We thus neglect the influence of colored noise due to either the higher-order dispersion or the inter-channel inelastic light scattering. This physical layer assumption remain sufficiently reasonable as long as the entire modulated bandwidth does not exceed the $(C+L)-$band [28], as in this work. Hence, the available path data rate estimated from the Shannon capacity is given by

$$\mathrm{C}_{\mathrm{path}} = 2R_S \cdot \log_2\left(1 + \frac{1}{\sum_i \mathrm{NSR}_i}\right), \tag{1}$$

where R_S is the symbol rate and NSR_i stands for the white noise-to-signal ratio (NSR) defined at the optimal launch power on $i^{\mathrm{th}}-$link. At the Nyquist rate, this is given by the following closed-form expression [1]

$$\mathrm{NSR}_i = N_i \sqrt[3]{\frac{2\sigma_{\mathrm{ASE}}^4 \alpha \gamma^2 L_{\mathrm{eff}}^2}{\pi |\beta_2| R_S^2} \ln\left(\frac{\pi^2 |\beta_2|}{\alpha} \cdot B^2\right)}, \tag{2}$$

[1]For the parameters given in Table 1, the NSR scales as: $\mathrm{NSR}_i \approx {}^{N_i}/_{405}$.

Table 1. Physical layer parameters

Parameter	Value	Units
Notional carrier wavelength (λ_0)	1550	nm
Symbol rate (R_S)	100	GBd
WDM channel spacing	100	GHz
Total modulated bandwidth (B)	10	THz
Loss coefficient (α)	0.2	dB/km
fiber GVD coefficient (β_2)	– 21.7	ps^2/km
Nonlinear coefficient (γ)	1.2	/W/km
Lumped amplifier spacing (L_s)	100	km
Lumped amplifier noise figure (NF)	4.5	dB

where N_i denotes the number of fiber spans on the $i^{\text{th}}-$link, σ^2_{ASE} is the amplified spontaneous emission (ASE) noise power, α is the fiber loss coefficient, β_2 is the fiber group-velocity dispersion (GVD) coefficient, γ is the fiber nonlinear coefficient, L_{eff} is the fiber effective length, B is the total modulated bandwidth, and $\ln(\cdot)$ denotes the natural logarithm. The overall variance of ASE noise arising from the lumped optical amplifiers at the end of each fiber span in a link is given by

$$\sigma^2_{\text{ASE}} = \left(e^{\alpha L_s} - 1\right) \cdot 10^{\frac{\text{NF}}{10}} \cdot \frac{hc}{\lambda_0} \cdot R_S\,, \tag{3}$$

where L_s is the fiber span length, NF stands for the lumped amplifier noise figure measured in [dB], λ_0 is the notional carrier wavelength, c is the speed of light in vacuum, and h is the Planck constant. The physical layer modeling parameters are shown in Table 1.

This physical layer model is used to pre-calculate the maximum capacity of each lightpath in the network, C_{path}. We then assume a demand with a given bit rate D, such that the remaining data rate for each lightpath R_{rem} after servicing a demand is given by

$$R_{\text{rem}} \triangleq C_{\text{path}} - D. \tag{4}$$

Thus, each lightpath can support multiple requests, if D is smaller than C_{path}. It should be emphasized that C_{path} is initialized assuming fully-loaded links, such that the Shannon capacity is never overestimated. Additionally, we note that this physical layer model, chosen for simplicity, could be modified to more closely resemble a given deployed system without loss of generality for the RWA method presented in this work.

4. REINFORCEMENT LEARNING RWA SOLUTION

A. Reinforcement learning theory

In RL, problems are represented as a finite MDP [16], consisting of an agent interacting with its environment at a series of time steps $1, 2, \ldots, t-1, t, t+1, \ldots$. We can think of the agent as a controller that learns through trial and error to perform a given task. This learning happens interactively through interaction with the environment, which in the context of RWA is a simulation of the optical network. The agent can choose from a set of available actions given its current observation of the state of the environment, defined as the observation space, and it receives a numerical reward for each action taken, describing how good the action was. In the case of RWA, an optimal action is one that maximizes the total number of services that can be supported by the network. Through many interactions with the environment, the agent learns a function known as the policy that describes the probability of each action being optimal given the current state. In the strict MDP formulation of RL used in this work, only the current state is inputted into this function. The user must design the environment, the observation space for the agent and the reward function. More quantitatively, the agent's goal is to learn an optimal policy Π^*, a functional mapping from the observed current state $S_t \in \mathcal{S}$ of the environment to the optimal action A^* ($\Pi^* : S \to A^*$). A numerical reward $R_{t+1} \in \mathcal{R} \subset \mathbb{R}$ is provided for each action $A_t \in \mathcal{A}$. Fig. 1 depicts the MDP for an RL agent solving RWA in an optical network. The agent aims to maximize the cumulative future reward G_t for timestep t, defined as follows

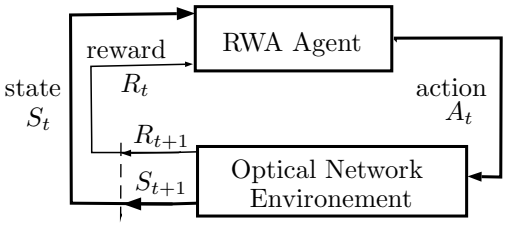


Fig. 1. Diagram of the MDP that defines the interactions between the RWA agent and the optical network.

$$G_t = \sum_{\tau=0}^{T-t-1} \kappa^\tau R_{t+\tau+1}\,, \tag{5}$$

where T denotes the total number of timesteps and $\kappa \in [0, 1)$ denotes the discount factor [16].

There are a range of different algorithms for finding an optimal policy, meaning a policy that maximizes Eq. (5). In this work, we utilize the proximal policy optimization (PPO) algorithm [29], in which the policy of the RL agent is represented by a NN such that it is possible to exploit the NNs ability to generalize to unseen states [7]. PPO is a state-of-the-art RL algorithm that has been observed to perform very well for a range of problems, including RWA [7]. We denote this NN parameterization of the policy by Π_θ. PPO is a policy gradient algorithm, meaning that the gradient of the expected cumulative reward of the policy is calculated and gradient ascent is used to update the NN parameters θ such that the expected cumulative reward is maximized. Said another way, PPO trains a NN representation of the policy Π_θ with the goal of finding the optimal policy Π^* defined above. PPO is also an on-policy algorithm. This means that the agent follows the current best estimate for the optimal policy when gathering samples. For a detailed description of PPO, we refer the reader to Ref. [29] and Ref. [7].

B. Invalid action masking

As discussed above, in PPO the policy is represented by a NN. This NN learns a mapping from the observation space defined above to the probability of each of the actions in the action space being the optimal choice, in terms of maximizing the cumulative future reward. Thus, these probabilities correspond to the output nodes of the NN (and the observation space corresponds to

the input nodes). In standard RL, the action corresponding to the output node with the highest probability is selected, as this is the action that is most likely to lead to maximization of the cumulative reward according to the current policy. In some problems, not all of the actions in the action space will be valid choices for every state of the environment and it is possible to know from the current state of the environment which actions are invalid. In the RWA problem for instance, not all of the lightpaths connecting the requested source and destination nodes will be able to support the request, either because they are blocked or they do not have sufficient capacity. Invalid action masking works by checking which actions are invalid and removing invalid actions from the action space. This is implemented by setting the output probabilities of the NN nodes for invalid actions to zero, then renormalizing the remaining probabilities and selecting the highest-probability valid action. In standard RL, the agent must learn through the reward signal how to avoid choosing invalid actions, which may be challenging. Thus, RL with invalid action masking is more efficient than standard RL as the agent does not have to learn to avoid invalid actions in this way.

C. Difficulties associated with long episodes

In general, the difficulty of using RL to solve a given problem increases as the size of the training episodes increases. This can be due to a number of factors, however in this problem there are two key reasons why this is the case. First, for longer episodes the agent has to make more decisions and it therefore becomes increasingly more difficult to quantify the importance of each decision in terms of reaching the final state of the episode, known as the credit assignment problem [15, 16]. Second, for longer episode sizes the number of full training episodes is smaller for a given fixed number of training timesteps. Thus, with longer episodes the agent experiences fewer full episodes for the same total training budget. As a result of these issues, we observed that standard RL was unable to learn a policy that outperformed baseline heuristics for the RWA problem with longer episodes compared to in the literature, i.e., with requests of a given fixed bit rate rather than connection requests.

D. Reinforcement learning agent design

In this work, the agent learns how to choose a route and a wavelength for a given service request in a series of requests, given the requested source and destination and the state of the network as input on each timestep. The key components of our RL model are as follows.

Environment The environment consists of a simulated fixed grid optical network, composed of nodes connected by bidirectional links. These links have a given maximum capacity, estimated using the physical layer model outlined in Section 3. Requests in the environment consist of 100 Gbps demands. If a demand is serviced on a given lightpath, the remaining capacity is calculated according to Eq. (4). The requests are generated by selecting two nodes randomly without replacement, where the probability of selecting each node is given by a traffic model. The salient features of the state of environment are the locations of allocated services on the network, meaning the link-channel combinations that they occupy, and the remaining capacity of each lightpath.

Observation space We consider a simple representation at the link-level, where links are equivalent to edges on a graph. Specifically, the observation space consists of the number of services on each link, as well as the source and destination nodes of the current request. This allows the agent to see the total link load across all channels on a given link. Using a link-level observation space as opposed to a path-level representation [7, 8, 10] allows the agent to see a whole-topology view of the network, at the cost of requiring the RL agent to learn the mapping from the link-level observation to a path-level action.

Episode An episode consists of a series of timesteps. In each training episode we begin with an empty network and sequentially receive non-expiring requests at a rate of one request per timestep, which the agent aims to service. This assumption of non-expiring requests is justified by the fact that requests are established for relatively long periods of time in currently deployed networks. Episodes terminate after a pre-determined number of timesteps (equals to the number received requests), after which the network state is reset to the initial empty state. The motivation for training the agent to perform RWA starting from an empty network state is as follows. Performing RWA from an initial empty state is the most challenging problem for the agent to solve, and thus it is expected that the proposed scheme could be applied to RWA in a brownfield scenario, meaning with a non-empty initial state. This is because starting from the empty state corresponds to the maximum episode length, which is the most challenging due to the credit assignment problem, as discussed above. For brownfield RWA, the episode size is shorter, and thus the difficulty of credit assignment is reduced. Moreover, it is noted that in RL the policy maps states to actions and that as the problem is formulated as an MDP, only the current state is required to make an optimal decision. Therefore, the optimal policy learned for RWA from the empty initial state is still optimal for brownfield RWA. Applying the presented RWA framework to brownfield RWA is part of the planned future work.

Action space with invalid action masking The action space consists choosing one of the $k \times N_{\text{ch}}$ unique lightpaths connecting the requested source and destination, where N_{ch} denotes the number of channels. We utilize domain knowledge to reduce the size of the action space using invalid action masking [30], which facilitates learning for problems in which the number of valid actions is not constant across all episodes. As more services begin to occupy the network, not all of the lightpaths connecting the desired source and destination nodes will be available, and therefore these lightpaths can be masked out such that the agent does not consider them. This corresponds to imparting some domain knowledge into the RL learning process, allowing the agent to focus on learning a successful allocation strategy given the lightpaths that will not result in immediate blocking.

As we consider non-expiring requests, the action space of the agent with invalid action masking becomes smaller as more services are provisioned in the network, as more of the lightpaths become blocked. We observed a significant increase in performance using invalid action masking compared to standard *tabula rasa* learning, in which the agent must learn through experience to not choose lightpaths that are already blocked.

Reward We used the following reward function, motivated by the shaped reward presented by Cicco et al. [(Eq. 21) 7]

$$r = \begin{cases} 1/L, & \text{if service accepted} \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

where L is the load on the chosen path, defined as the sum of the services allocated on each of the constituent links in the path.

L was normalized to a maximum value of 1, such that the value of the reward returned was within a reasonable range. Thus, this reward is intended to bias the agent towards choosing paths with low loading. We observed an increase in the efficacy of the agent as compared to the simplistic rewards used in the literature [7, 10, 12]. Specifically, this reward is defined as $+1$ for a successfully serviced request and -1 for a rejected request. We also tried a similar reward function of $+1$ for an accepted request and 0 for a rejected request which has been used in the literature also [8, 14]. Again, the reward defined in Eq. (6) yielded favorable performance.

5. TRAINING AND EVALUATION SET-UP

A. Proposed training methodology for problems with long episodes

The choice to use a simplified version of the RWA problem for training the RL agent was motivated by the credit assignment problem, caused by the large episode sizes encountered in our formulation of RWA. Specifically, training the agent is difficult with large episode sizes as it becomes difficult for the agent to determine the importance of each decision made, as outlined in Section 4. Therefore, we train the agent on a simplified version of the problem, in which the episode size is reduced. This is achieved by multiplying the capacity of each lightpath in the network by a scale factor SF:

$$C_{\text{path}} \coloneqq C_{\text{path}} \times \text{SF}. \tag{7}$$

As a result of this, the number of 100 Gbps services that the network can support is reduced, and thus the episode size can be reduced. The episode size was reduced by the same factor SF. The agents were trained on this scaled problem and then evaluated on the full target problem, i.e. with SF $= 1$. We found that a value of SF $= 0.2$ worked well for both the NSFNET and COST-239 topologies. This suggests that a highly effective policy can be learned on the scaled problem and applied directly to the target problem. We observed an increase in performance of the RL agent using this training mode as compared to regular training on the target problem. This is due to a reduction in the difficulty of credit assignment, aiding the agent to learn an effective RWA solution. A similar demonstration of policy generalization was shown in Ref. [31], where an RL agent learned a policy for resource allocation in datacenters - another graph-based allocation problem (related to the RWA problem considered here). Here the agent was evaluated on graphs that were 100 times larger than those it was trained on in terms of the number of nodes. Whilst this is different to the scaling applied in Eq. (7), it demonstrates that training on a smaller problem can often lead to a higher efficacy RL solution than training directly on the target problem.

B. Network topologies

In this study, we consider two core network topologies, namely the NSFNET topology and the COST–239 topology [32], shown in Figure 2 and Figure 3, respectively. All link lengths are shown in km and have been rounded to the nearest integer multiple of 100 km, as we assume a fixed span length of 100 km at the physical layer implementation. The 11-node COST-239 topology has an average node degree of 4.7 and mean link length of 581 km compared to 3.1 and 945 km for 14-node NSFNET.

C. Simulation Environment

The network simulation is built using the open source Optical RL Gym library [14] and the agents are trained using implementations of PPO with and without invalid action masking provided by the Stable Baselines 3 library [33], referred to as MaskablePPO and PPO, respectively. In this work, the RL agent is trained on the NSFNET (Fig. 2) topology using the simplified training process outlined above for a total of 10^7 timesteps. With the chosen value of SF $= 0.2$, this corresponds to 5000 episodes of 2000 timesteps each (with one request per timestep). For the COST-239 topology, training was performed also with SF $= 0.2$ for a total of 10^7 timesteps, meaning 5000 episodes of size 4000 each. The target problem for NSFNET and COST-239 consists of 10^4 and 2×10^4 timesteps respectively (with SF $= 1.0$). This difference is due to the fact that COST-239 has a higher capacity than NSFNET, due to its higher average node degree and shorter average link length. Training was performed for both topologies with the following hyperparameters: discount factor $\kappa = 0.99$, learning rate of 1.57×10^{-5}, batch size equal to 16 and a network architecture of 2 layers of 128 neurons. All other parameters are equal to the defaults in Stable Baselines 3 MaskablePPO and PPO. We use a fixed bit rate of 100 Gbps.

During evaluation the same requests are given to each algorithm per episode. Lightpaths are modeled as having a given capacity Eq. (1), meaning that an existing lightpath can be used to service multiple requests between the same two nodes as long as there is sufficient spare capacity and wavelength continuity is obeyed. $k = 5$ for both the agent and the heuristics, meaning that both can choose up to the $5^{\text{th}}-$shortest path.

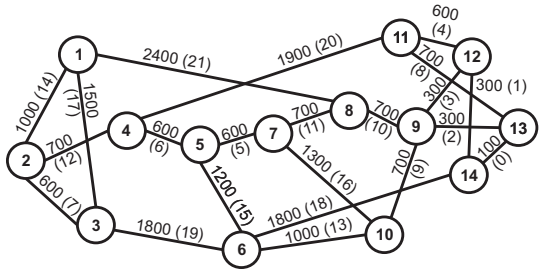


Fig. 2. 14−node NSFNET topology. Link lengths are given in km, and link IDs are shown in brackets and have been ordered from shortest to longest.

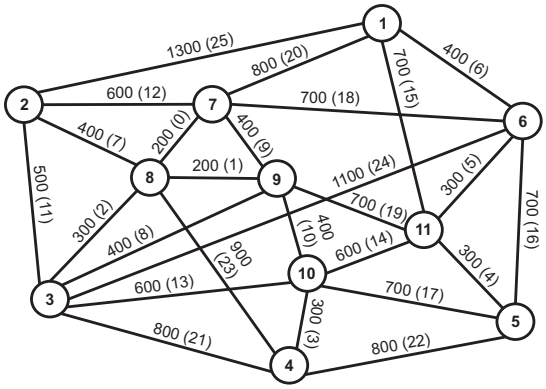


Fig. 3. 11−node COST-239 topology. Link lengths are given in km, and link IDs are shown in brackets and have been ordered from shortest to longest.

D. Heuristic baselines

We benchmark the performance of our RL solution against three state-of-the-art RWA heuristics: $k-$shortest path first fit,

k-shortest path most-used (kSP-MU) and first fit $k-$shortest path (FF-kSP) [19]. kSP-FF searches for a lightpath that can support the current request, starting with the shortest path and searching each channel sequentially until a valid lightpath is found. If a lightpath is not found for the shortest channel, the second-shortest path is searched and so on. kSP-MU also searches each channel in order of length, allocating the request to the most-used wavelength in the network at the current time. This heuristic is motivated by reducing the number of channels used in order to reduce congestion. Finally, FF-kSP starts with the first channel slot and searches each of the shortest paths in order of length to find a lightpath that can support the current request. Thus, FF-kSP also aims to reduce the number of channels used via wavelength packing in an attempt to reduce network congestion [19].

6. RESULTS FOR UNIFORM TRAFFIC

A. Uniform traffic model

We consider a traffic model with non-expiring requests similar to Vincent et al. [19]. Bidirectional symmetric traffic is assumed and we consider the uniform-all-to-all model [34]. For a network graph $\mathscr{G} = (\mathcal{V}, \mathcal{E})$ with a set of nodes $\mathcal{V} \triangleq \{v_1, v_2, \ldots, v_N\}$ and a set of edges $\mathcal{E}$, source nodes $v_i \in \mathcal{V}$ and destination nodes $v_j \in \mathcal{V}$ $(i \neq j)$; the uniform traffic matrix $\widehat{T}_{\text{unif}}$ is defined as [34]

$$\widehat{T}_{\text{unif}}: \quad \forall \left\{v_i, v_j\right\} \in \mathcal{V}: \quad T_{ij} = \frac{1}{N\,(N-1)}, \tag{8}$$

where $N \triangleq |\mathcal{V}|$ is the total number of nodes in a given network.

B. Impact of action masking

Invalid action masking allows us to restrict the action space of the RL agent to only viable lightpaths, such that the agent can focus on learning an effective RWA strategy, rather than learning through the reward signal how to identify lightpaths that are currently blocked. Here we investigate the performance benefit of invalid action masking as compared to a standard fixed-size action space.

We benchmark using the NFSNET topology, shown in Figure 2, for uniform traffic Eq. (8). The results shown were recorded across 100 evaluation episodes, where the traffic requests are drawn a uniform traffic matrix and the episode starts with an empty network and ends after 10^4 service requests.

Figure 4 depicts the learning curves for RL agents trained with and without invalid action masking. The moving average of the number of accepted services calculated with a sliding window of size 5 is shown, in order to show the general trend. These models are trained using a uniform traffic matrix for 10^7 timesteps using the simplified training process outlined in Section 5. We found empirically that a scale factor SF $= 0.2$ had a high efficacy. Thus, using this technique the episode length is reduced by a factor of 5 and the number of full training episodes is 5 times larger for the same total number of timesteps. This resulted in an increase in the efficacy of the RL agent of 1.8% relative to training directly on the target problem in terms of the number of serviced requests. As we trained using 10 parallel computational processes [33], we show typical learning curves for one process for the entire 10^7 timesteps, meaning 500 episodes per process. Thus, Figure 4 is representative of the whole training run. We can see that RL with invalid action masking is able to service a significantly higher number of requests compared to without masking. We note that the starting average

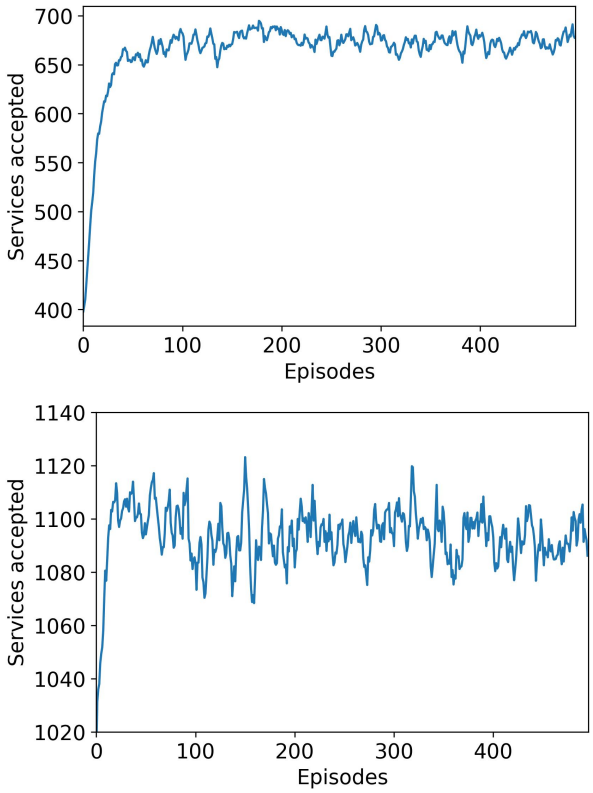


Fig. 4. Training curves for RL agents without (top) and with (bottom) invalid action masking for simplified RWA problem with SF $= 0.2$. The moving average calculated with a step-size of 5 episodes is shown. As training was performed using 10 parallel processes, learning curves are shown for one typical individual process. The training was for a total of 10M timesteps, corresponding to 5000 episodes, with 500 episodes per process, with uniformly-distributed traffic on the NFSNET topology.

value for the accepted services is much lower for the standard RL agent as compared to the agent with invalid action masking - approximately 400 as compared to 1020. This is because without invalid action masking, the agent is able to choose lightpaths that are currently blocked. Thus, the agent must learn how to identify lightpaths that are currently blocked from the observation space and the reward signal. In the invalid action masking case, the agent can instantly start learning an effective strategy for choosing from the available (non-blocked) lightpaths. This makes the learning process more efficient, allowing the agent to learn a solution with higher efficacy. Moreover, use of invalid action masking reduces the size of the action space for the majority of the episode. Once services start being established in the network, the number of lightpaths that are able to support the current request is reduced and thus more actions will be masked. This will reduce the size of the action space and thus the complexity of the RL formulation, making it easier for the RL agent to learn an effective policy. Moreover, we also note that the learning curves are observed to converge rapidly for this simplified case both with and without action masking, plateauing after a relatively small number of timesteps in both cases. This is due to the fact that we have simplified the training problem significantly, such that the agent can rapidly learn an

effective RWA solution. Investigating the minimum number of training timesteps required to achieve a highly performant solution forms part of the planned future work.

In Figure 5 we compare the performance of the RL agent with and without invalid action masking over 100 evaluation episodes with uniformly-distributed traffic on the NSFNET topology. It is important to note that while the agents were trained on a scaled-down version of the RWA problem with SF $= 0.2$, they were evaluated on the target problem with an episode length of 10^4 requests. We also include results for random RWA, where we choose the path and channel at random, and the first-fit heuristics for reference. Again, we observe a significant increase in performance as a result of the inclusion of invalid action masking, with an increase of 222% in the mean number of services provisioned as compared to the RL agent without masking. We saw similarly poor performance without invalid action masking for the COST-239 topology considered, with an increase of 100% in the number of accepted services with invalid action masking as compared to without.

In the literature standard PPO algorithms have achieved similar or greater performance than standard first-fit heuristics such as kSP-FF [7, 8, 10, 12]. However, we did not observe this due to a key difference between our simulated network environment and those proposed in the literature, as outlined above. Specifically, in the literature of RL-driven RWA in fixed grid optical networks, the network environment has been set up such that the agent has been tasked with servicing a series of connection requests [7]. Thus, the channels can only accept one service. However, we model lightpaths that can support multiple services, as long as they have sufficient remaining capacity as calculated using the GN model. This poses a more difficult problem for the agent, for a number of reasons. Firstly, as outlined above, credit assignment becomes increasingly more difficult as the length of episodes increases. Furthermore, the agent must also learn how to effectively re-use lightpaths as well as how to allocate new ones. This is not required when servicing a series of connection requests, as in the literature. Additionally, this means that the episodes are longer as more services can be supported on the network as compared to a more simplistic network with single-occupancy channels, increasing the difficulty of the problem as fewer training episodes are available in the same wall-clock training time with fixed compute resources. In order to address these problems, we found that invalid action masking, i.e. removing lightpaths that cannot accept the current request from the action space, and a training methodology in which we trained on a scaled-down version of the RWA problem was required.

Additionally, the observation space we have used is relatively simplistic and a more complex space may result in better performance without invalid action masking. A rigorous study of the effects of different observation spaces on the performance of RL both with and without masking forms part of the planned future work. We also performed hyperparameter tuning with each of the models tested, performing parameter sweeps of the learning rate, discount factor, batch size and network architecture. However, further exploration may yield hyperparameters with higher efficacy both with and without action masking.

C. Benchmarking with Heuristics

Figure 6 shows boxplots of the number of accepted services across the 100 evaluation episodes for the RL agent with invalid action masking and the heuristics for uniform traffic for NSFNET. The agent is trained and evaluated on uniformly-distributed traffic in this case, and key statistical metrics are summarized in Table 2 for reference. We can see that for the uniform traffic case, the RL agent outperforms all heuristics in terms of the mean, median, minimum and maximum number of services accepted. Compared to the best-performing heuristic for this case, this translates to an increase of 184, 183 and 195 services for

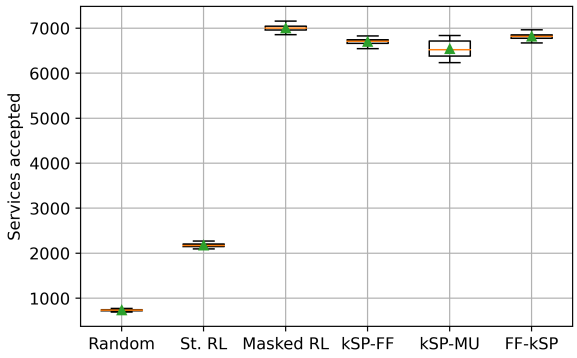


Fig. 5. Boxplot showing the number of services accepted across 100 evaluation episodes for the RL agent with (Masked RL) and without (St. RL) invalid action masking. The performance of a random agent and the first-fit heuristics are also shown for reference. Each agent was trained for 5000 episodes on the NSFNET topology, corresponding to 10^7 timesteps, with a uniform traffic distribution. A simplified version of the target problem with a smaller episode size was used for the training, as outlined above.

Table 2. Evaluation statistics for uniform traffic on NSFNET

	kSP-FF	kSP-MU	FF-kSP	RL
Median	6710	6525	6818	**7002**
Mean	6701	6543	6820	**7002**
Min	6545	6234	6674	**6857**
Max	6831	6841	6964	**7159**
SD	**55**	175	63	59
IQR	80	332	**75**	83

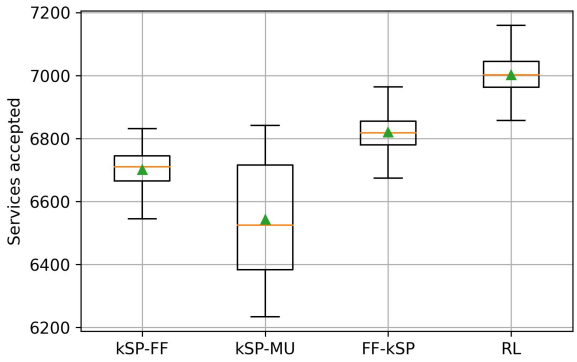


Fig. 6. Boxplots showing the number of services accepted across 100 evaluation episodes for the RL agent and heuristics for uniformly-distributed traffic on NSFNET.

Table 3. Evaluation statistics for uniform traffic on COST-239

	kSP-FF	kSP-MU	FF-kSP	RL
Median	15156	14208	14624	**15279**
Mean	15156	14170	14624	**15283**
Min	14968	12921	14308	**15004**
Max	15333	15345	14865	**15549**
SD	**80**	1015	126	110
IQR	**119**	2012	163	141

the median, worst-case and best-case respectively. Additionally, the agent has a similar interquartile range (IQR) to kSP-FF and FF-kSP, indicating a robust solution. This improvement should be understood in the context of the novelty of this paper. These results indicate for the first time the applicability of RL to RWA in fixed grid networks for 100 Gbps demands, i.e. with re-use of extant lightpaths with sufficient capacity. The improvement of a median of 184 services corresponds to an increased throughput of 18.4 Tbps, which may be of significant value to network operators. Additionally, we note that the best performing heuristic for uniform traffic, FF-kSP, is not the best performing for population-based traffic. On the other hand, the agent is consistently the highest performing across the two traffic types. Thus, the agent affords the operator a flexibility benefit with respect to different traffic distributions.

Moreover, the Friedman non-parametric test [35] suggests RL outperforms the heuristics with statistical significance of 99.5%. Additionally, we performed an evaluation of the RWA run time of the trained agent compared to the heuristics and demonstrated that the run time of the RL agent is of the same order as the heuristics. As the costs of training time are mitigated by the ability to pre-train the agent before performing RWA, this shows the applicability of our RL solution in terms of computational requirements to real world systems.

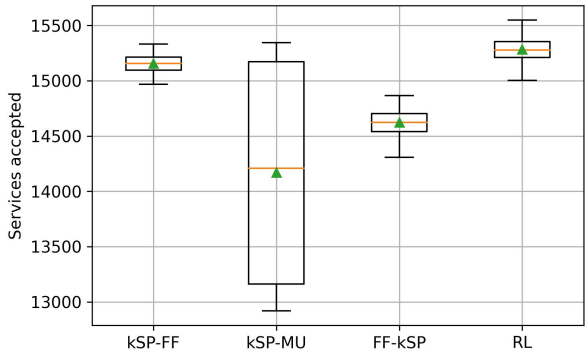


Fig. 7. Boxplots showing the number of services accepted across 100 evaluation episodes for the RL agent and heuristics for uniformly-distributed traffic on COST-239.

For the COST-239 topology, training was also performed using the methodology outlined in Section 5 on a scaled-down version of the RWA problem with SF $= 0.2$ for 10^7 timesteps and the learned policy was evaluated on the full realistic RWA problem, i.e. with SF $= 1$. As COST-239 has a higher average node degree, more total links and a shorter average link length than NSFNET, the capacity of this network is higher. For instance, we found that all RWA algorithms tested were able to service 10^4 requests without blocking. Thus, we increased the episode length to 2×10^4. Hence, the issues outlined in Section 4 related to long episodes are worse for COST-239 as compared to NSFNET. The performance relative to the heuristics for uniform traffic is shown in Figure 7 for 100 evaluation episodes and summary statistics are provided in Table 3. The RL agent has the highest median, mean, minimum and maximum performance, as for NSFNET. The SD and IQR are also comparable to kSP-FF and FF-kSP. As for NSFNET, we observed that kSP-MU has a large variance in performance, perhaps due to sensitivity to the first few services in the network in terms of determining the most-used wavelength. Investigating this and other more advanced heuristics and comparison with upper bounding global solutions such as integer linear programs forms part of the planned future work.

These results should be interpreted in the context of the novelty of this work. Specifically, this demonstrates the feasibility of RL for more detailed RWA problems involving demands of a fixed bit rate in core networks, i.e. for problems with long episode sizes. Crucially, we observed that standard RL is unable to approach baseline heuristics for such problems using methods similar to those deployed on connection request RWA problems in the literature, due to the significant increase in episode size. Thus, we introduced invalid action masking and a simplified training environment to improve the efficacy of the agent in these scenarios.

7. GENERALIZATION TO REALISTIC TRAFFIC

A. Population-based traffic model

In order to investigate the generalizability of this RL model, we evaluate the RL agent trained on uniformly-distributed traffic presented in Section 6 on a realistic non-uniform traffic distribution not seen during training, generated by considering population of major US states obtained from the 2020 US census [36]. Let S and D be the discrete random variables denoting the source and the destination, respectively. Possible values each can take are $1, 2, \ldots, i, \ldots, k$ with k being the number of nodes. If r_i is the number of residents for the $i^{\text{th}}-$node, then we assume the probability of selecting the source is the population of the source as a fraction of the total population such that

$$\mathbb{P}\left[S = i\right] = \frac{r_i}{\sum_k r_k}, \tag{9}$$

and likewise the conditional probability of the destination is the population of the destination as a fraction of the remaining population (i.e., excluding the source population) such that

$$\mathbb{P}\left[D = j \,|\, S = i\right] = \frac{r_j}{\sum_{k \neq i} r_k}, \tag{10}$$

and likewise by symmetry

$$\mathbb{P}\left[S = j\right] = \frac{r_j}{\sum_k r_k}, \tag{11}$$

$$\mathbb{P}\left[D = i \,|\, S = j\right] = \frac{r_i}{\sum_{k \neq j} r_k}. \tag{12}$$

Table 4. Non-uniform population-based traffic matrix for NSFNET given by Eq. (15), where all actual values of probabilities are multiplied by 10^4.

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	47	76	9	17	94	5	39	40	32	30	63	27	2
2	47	0	157	20	36	195	11	80	82	67	63	131	55	4
3	76	157	0	31	57	312	19	129	131	107	101	208	89	7
4	9	20	31	0	7	39	2	16	16	13	13	26	11	1
5	17	36	57	7	0	71	4	28	30	23	23	47	20	2
6	94	195	312	39	71	0	23	160	164	134	126	261	111	8
7	5	11	19	2	4	23	0	10	10	8	8	16	7	1
8	39	80	129	16	28	160	10	0	67	55	52	108	45	2
9	40	82	131	16	30	164	10	67	0	56	53	110	46	4
10	32	67	107	13	23	134	8	55	56	0	43	90	38	2
11	30	63	101	13	23	126	8	52	53	43	0	84	36	2
12	63	131	208	26	47	261	16	108	110	90	84	0	74	5
13	27	55	89	11	20	111	7	45	46	38	36	74	0	2
14	2	4	7	1	2	8	1	2	4	2	2	5	2	0

Hence, for an unordered pair (i, j) of source and destination nodes

$$\mathbb{P}[i,j] = \mathbb{P}[D = j \,|\, S = i] \cdot \mathbb{P}[S = i] + \mathbb{P}[D = i \,|\, S = j] \cdot \mathbb{P}[S = j]\,. \tag{13}$$

However, for a traffic matrix with elements T_{ij}

$$\mathbb{P}[i,j] \triangleq T_{ij} + T_{ji}\,. \tag{14}$$

Therefore, assuming an undirected graph with a symmetric traffic matrix requires $T_{ij} = T_{ji}$, and hence

$$T_{ij} = \frac{1}{2}\left(\frac{r_i}{\sum_k r_k}\frac{r_j}{\sum_{k\neq i} r_k} + \frac{r_j}{\sum_k r_k}\frac{r_i}{\sum_{k\neq j} r_k}\right)\,. \tag{15}$$

The source and destination nodes of each request are generated randomly using the node request probabilities for the population-based traffic matrix given by Eq. (15). We show the values of the traffic matrices for NSFNET and COST-239 in Table 4 and Table 5 respectively, where the probabilities have been rounded and multiplied by 10^4. An additional colormap visualization of the traffic matrices for NSFNET and COST-239 are provided in Figure 8 and Figure 9.

B. Benchmarking with heuristics

We first consider the NSFNET topology. Boxplots showing the generalization of the RL model to the unseen population-based non-uniform traffic distribution and a summary of key statistics are presented in Figure 10 and Table 6 respectively. For this case, the RL agent achieves the highest mean, median, best-case and worst-case number of accepted services. Specifically, the RL agent serviced an extra 147, 170 and 144 requests for the median, best-case and worst-case respectively, compared to the best-performing heuristic kSP-FF. Also, the agent achieved the second-lowest standard deviation (SD) and IQR, indicating that the RL solution is robust across evaluation episodes. As with the uniformly-distributed traffic results, this improvement should be understood in the context of the novelty of this paper. Namely, these results indicate for the first time the applicability of RL to

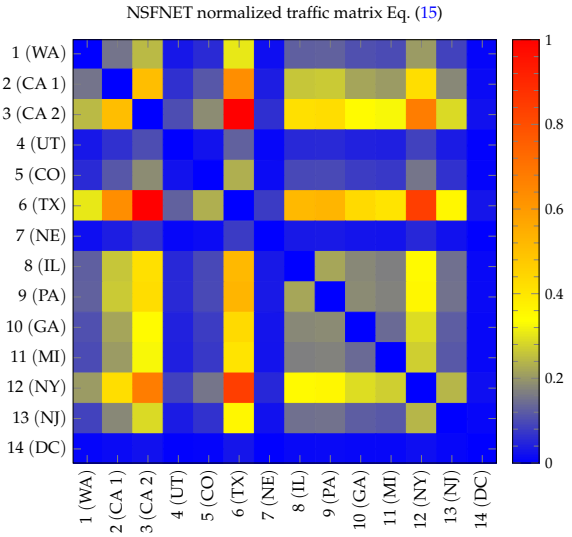


Fig. 8. Colormap visualization of non-uniform population-based normalized traffic matrix for NSFNET.

Table 5. Non-uniform population-based traffic matrix for COST-239 given by Eq. (15), where all actual values of probabilities are multiplied by 10^4.

ID	1	2	3	4	5	6	7	8	9	10	11
1	0	177	34	21	30	60	14	19	10	5	20
2	177	0	480	303	432	847	200	267	142	91	283
3	34	480	0	57	83	164	38	51	27	17	54
4	21	303	57	0	52	103	23	32	17	11	34
5	30	432	83	52	0	147	34	46	23	16	49
6	60	847	164	103	147	0	68	91	47	31	95
7	14	200	38	23	34	68	0	21	11	7	22
8	19	267	51	32	46	91	21	0	15	10	30
9	10	142	27	17	23	47	11	15	0	5	16
10	5	91	17	11	16	31	7	10	5	0	10
11	20	283	54	34	49	95	22	30	16	10	0

RWA in fixed grid networks for 100 Gbps demands, i.e. with re-use of extant lightpaths with sufficient capacity. The improvement of a median of 147 services corresponds to an increased throughput of 14.7 Tbps, which may be of significant value to network operators. Also, generalization of the learned policy to a different traffic matrix to that used in training is shown by this performance relative to the heuristics. Additionally, we can see that the performance of the heuristics does not generalize well across different traffic matrices. FF-kSP is the best heuristic for uniform traffic but performs worse than kSP-FF for the population-based traffic, suggesting that a wavelength packing strategy is not optimal for this case. However, the RL agent is able to learn a generalizable policy from the uniform traffic distribution during training, allowing it to perform well for a different, non-uniform traffic distribution without retraining. Therefore, the RL agent affords the operator a flexibility advantage over the heuristics, which need to be hand-tuned for each problem. This flexibility is one of the major advantages of RL-driven solutions.

Moreover, the Friedman non-parametric test [35] suggests RL

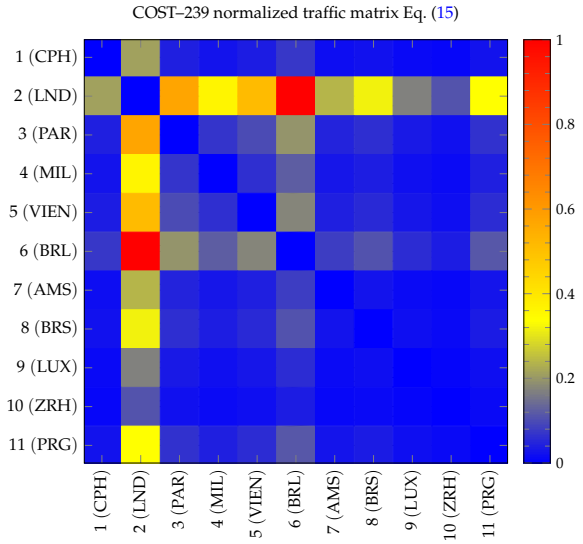


Fig. 9. Colormap visualization of non-uniform population-based normalized traffic matrix for COST–239.

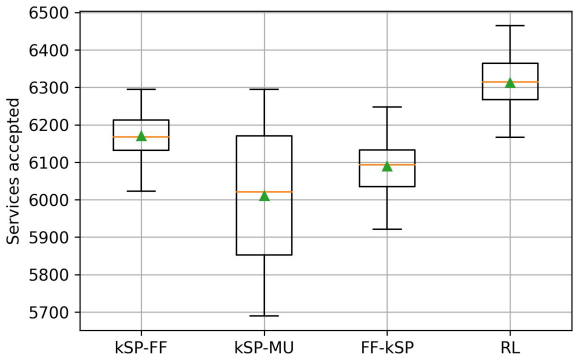


Fig. 10. Boxplots showing the number of services accepted across 100 evaluation episodes for the RL agent and heuristics for the non-uniform traffic distribution, indicating the ability of the RL model to generalize to a realistic traffic distribution on NSFNET.

outperforms the heuristics with statistical significance of 99.5%. Also, we evaluate the RWA run time of the trained agent for 100 evaluation episodes consisting of 10^4 sequential requests. For NFSNET under uniform traffic, the mean run times are 33.8, 37.8 and 54.2 seconds for the kSP-FF, FF-kSP and kSP-MU heuristics respectively, while the RL agent achieved a mean time of 47.3 seconds. Thus, kSP-FF is the fastest, followed by FF-kSP, the RL agent and then kSP-MU. Similar results are observed for population-based traffic matrix with means of 36.6, 59.5, 42.1 and 51.8 seconds or kSP-FF, kSP-MU, FF-kSP and RL respectively. Results for COST-239 follow similarly, that the RL agent run time is of the same order as the heuristics, showing the strong potential for applicability of our RL solution to real world systems.

Table 6. Evaluation statistics for non-uniform traffic on NSFNET

	kSP-FF	kSP-MU	FF-kSP	RL
Median	6168	6021	6093	**6315**
Mean	6171	6011	6090	**6313**
Min	6023	5690	5921	**6167**
Max	6295	6295	6248	**6465**
SD	**56**	170	76	66
IQR	**81**	318	98	96

For the COST-239 topology, the RL models presented in Figure 7 were evaluated for the population-based non-uniform traffic distribution defined above. Thus, we evaluated the generalization of the policy learned on uniform traffic to a realistic non-uniform traffic distribution. Boxplots showing the evaluation results for 100 episodes for non-uniform traffic are shown in Figure 11 and summary statistics are provided in Table 7. As for NSFNET, the agent achieves the best mean, median, minimum and maximum performance compared to the heuristics. Specifically, the RL agent serviced an extra 131, 31 and 240 requests for the median, best-case and worst-case respectively, compared to the best-performing heuristic FF-kSP. Also, the agent achieves the second-lowest SD and lowest IQR, indicating a robust solution. As with the other results presented in this paper, this relative improvement demonstrates that RL is feasible for RWA problems with longer episode sizes that are typical for servicing typical 100 Gbps demands in fixed-grid core optical networks. However, we observed that the use of invalid action masking and training on a simplified version of the problem was required to obtain this feasibility. Moreover, as with NSFNET, the best-performing heuristic for uniform traffic is not the optimal heuristic for the non-uniform case, whereas the RL agent is the highest performing in both cases. This indicates that the policy learned by the agent has the ability to generalize to an unknown traffic distribution for COST-239, as well as for NSFNET. The relative magnitude of the increase in performance relative to heuristic baselines is lower for COST-239 as compared to NSFNET, indicating some topology sensitivity. However, this may be due to the increased capacity of COST-239 and the associated increased episode length required to fill the network. In future we plan to investigate the use of a graph NN as part of the representation of the policy within the RL framework, which would allow learning from graph-scale features, thus reducing topology sensitivity.

Table 7. Evaluation statistics for non-uniform traffic on COST–239

	kSP-FF	kSP-MU	FF-kSP	RL
Median	11610	11560	11975	**12106**
Mean	11611	11542	11990	**12106**
Min	11455	11166	11736	**11976**
Max	11775	11864	12198	**12229**
SD	**65**	140	102	70
IQR	84	227	132	**74**

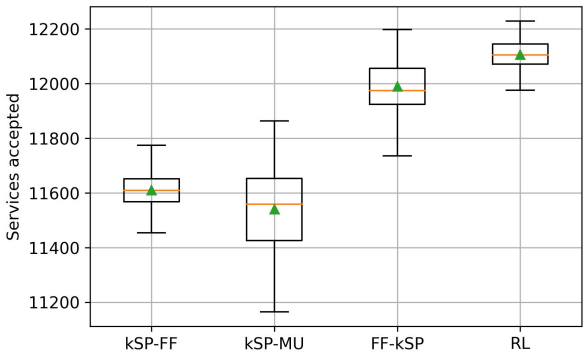


Fig. 11. Boxplots showing the number of services accepted across 100 evaluation episodes for the RL agent and heuristics for the non-uniform traffic distribution, indicating the ability of the RL model to generalize to a realistic traffic distribution on COST-239.

8. INTERPRETATION OF LEARNED RL POLICY

In order to interpret the policy that has been learned by the RL agent, and thus infer how it is able to outperform the heuristics, we visualize how the distribution of services varies during the evaluation episodes. This is done for uniform traffic in order to simplify the interpretation. Specifically, we record the distribution of services after 30% and 60% of the episode, corresponding to 3000 and 6000 services provisioned for NSFNET. This is so that we capture the distribution both early in the episode and near saturation. Also, we average the results across the 100 evaluation runs. Corresponding results for COST-239 are omitted due to space limitations, however they show similar trends to those for NSFNET.

For NSFNET, the service distributions after 3000 and 6000 services provisioned averaged over the 100 evaluation episodes for the RL agent, kSP-FF, kSP-MU and FF-kSP heuristics are shown in Figure 12. We show the number of services allocated to each link and channel for each RWA algorithm, in order to infer how the learned policy of the RL agent differs from the sequential heuristics. The link IDs are as denoted in the NSFNET topology (see Figure 2) and have been ordered from the shortest at 0 to the longest at 21. These link IDs are labeled in Figure 2.

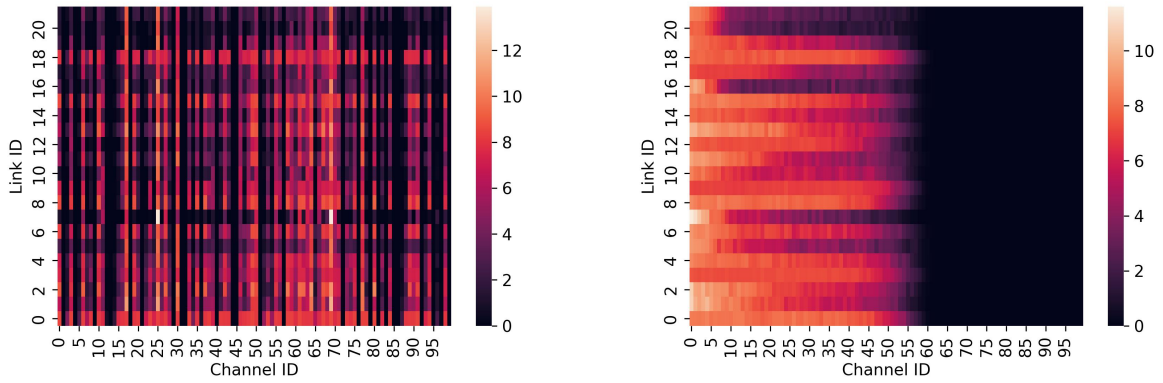
First, we can see that the RL policy is more complex than that of the sequential heuristics, with no clear bias for choice of channels. This lack of bias is due to the fact that we have modeled the channels as having an equivalent capacity, meaning that the agent has no bias towards the channel ID. On the other hand, the heuristics have a clear sequential strategy with respect to the channels as expected, which is particularly evident from the service distributions after 3000 services. For instance, for FF-kSP we can see the wavelength packing strategy in effect, leading to a more even loading across the links and fewer channels being utilized early in the episode. Contrastingly, kSP-FF shows a tendency to spread services across more channels as it is biased towards choosing shorter links. Additionally, due to this uniform spread of channels, the RL agent service distribution after 6000 services shows a tendency to leave more lightly-loaded links compared to the best-performing heuristic FF-kSP.

To further interpret the learned policy of the RL agent, we consider the distribution across the channels for the RL agent and heuristics. For NSFNET, the number of services assigned to each channel ID, averaged across all the links in the network and all 100 evaluation episodes is shown after 3000 services provisioned and 6000 services provisioned in Figures 13 and 14 respectively for the RL agent and heuristics. Figure 13 shows that FF-kSP uses the fewest channels of the heuristics early in the episode, whereas kSP-MU and kSP-FF use a similar number of channels at the early stages of the episode. Later in the episode at 6000 services, we can see that kSP-FF is the most biased towards lower channel IDs due to its first-fit wavelength selection policy, whereas kSP-MU and FF-kSP show a more even distribution across channels. Additionally, we can see from both Figure 13 and Figure 14 that the RL agent is spreading the services relatively uniformly across the channels, with no clear bias towards channel ID.

We also consider the distribution of services across the links, averaged across all the channels and the 100 evaluation runs. In Figure 15 the averaged distribution of services across links after 3000 services provisioned is compared for the RL agent and the heuristics for NSFNET. Here the link IDs have been ordered by length, from the shortest at ID 0 to the longest at ID 21. As expected, there is a general trend of more services on shorter links, as these links have a higher maximum capacity. Also, we can see that the RL agent distribution differs fairly consistently from the heuristics. For the two shortest path heuristics, kSP-FF and kSP-MU, the distribution is similar as expected, however FF-kSP prioritizes wavelength packing over shortest paths and thus it has a different distribution for the majority of links. Thus, the RL agent appears to be following a different strategy from the heuristics with respect to the location of the services on the links. For the distribution after 6000 services shown in Figure 16 the situation is similar, with the RL agent following a different strategy to the two heuristics for many of the links. In order to quantify this, we consider the mean absolute error (MAE) of the difference in link distribution between the RL agent and each heuristic. After 6000 services, the MAE values are 0.41, 0.42 and 0.46 for kSP-FF, kSP-MU and FF-kSP respectively. Thus, the RL policy is most similar to kSP-FF and least similar to FF-kSP with respect to the distribution of services across links.

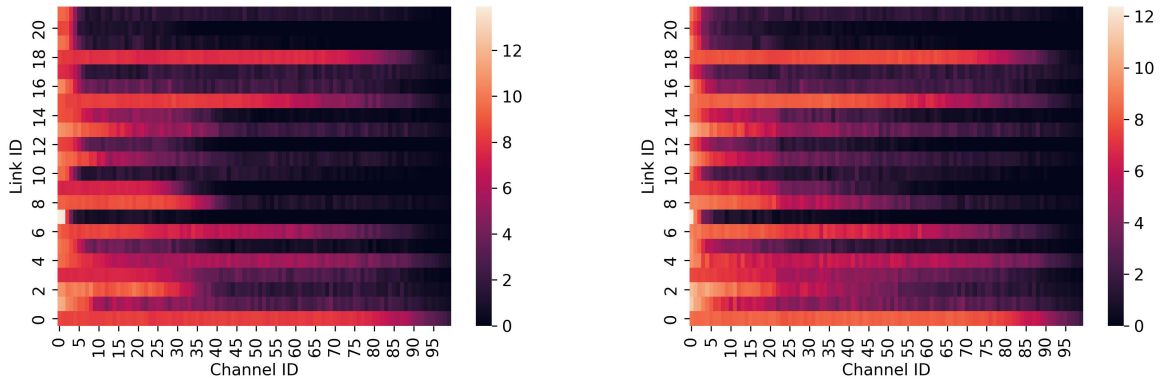
9. CONCLUSIONS AND FUTURE WORK

This study evaluated the efficacy of RL with invalid action masking for solving the RWA problem in fixed grid WDM optical networks. We model a physical layer that enables reuse of lightpaths with sufficient capacity, meaning that the RL agent has to learn how to reuse lightpaths as well as allocate new ones. This constitutes a more complex problem compared to the binary channel or frequency slot occupancy as commonly considered in the literature [7, 10, 12], particularly due to increased difficulty of credit assignment. The proposed approach to solve this problem constitutes a form of domain knowledge-informed RL, in which we constrain the action space such that the agent can only choose lightpaths that can support the current request. Additionally, we propose a training methodology in which the agent is trained on a simplified version of the RWA problem and the learned policy is applied to the target RWA problem. This was found to improve the efficacy of the agent compared to training directly on the target problem. We compare the performance of this RL solution with the performance without action masking, demonstrating significantly improved solution quality in terms of the total number of requests serviced. Additionally, comparison with the state-of-the art RWA heuristic approaches kSP-FF, kSP-MU and FF-kSP for uniformly-distributed traffic shows that the proposed knowledge-informed RL agent outperforms the



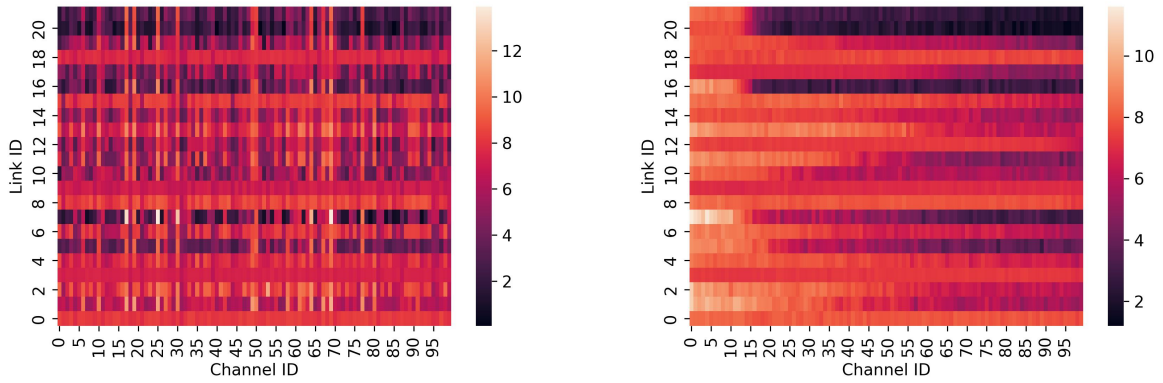
(a) RL service distribution after 3000 services.

(b) FF-kSP service distribution after 3000 services.



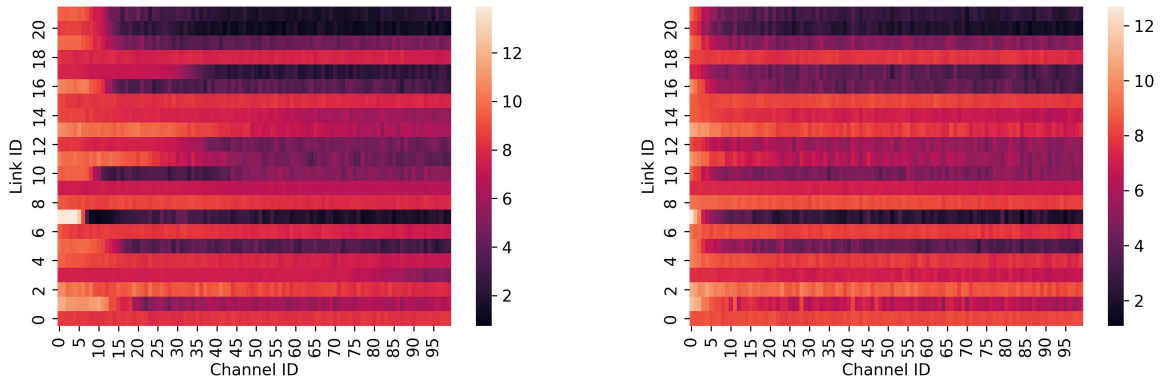
(c) kSP-FF service distribution after 3000 services.

(d) kSP-MU service distribution after 3000 services.



(e) RL service distribution after 6000 services.

(f) FF-kSP service distribution after 6000 services.



(g) kSP-FF service distribution after 6000 services.

(h) kSP-MU service distribution after 6000 services.

Fig. 12. Comparison of the distribution of services for the RL agent and heuristics after 3000 and 6000 services provisioned for uniformly-distributed traffic, averaged over 100 evaluation runs.

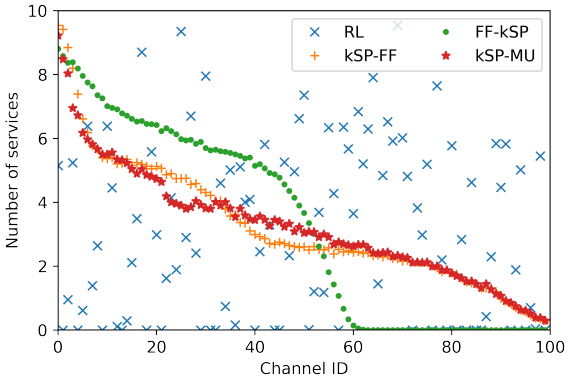


Fig. 13. Channel service distribution after 3000 services provisioned, averaged over all the links in the network and 100 evaluation runs, for the RL agent solution and heuristics on NSFNET. The RL agent distributes channels more uniformly than the heuristics, with no clear bias towards channel ID.

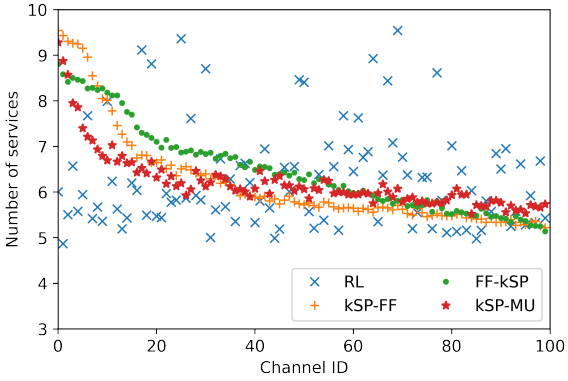


Fig. 14. Channel service distribution after 6000 services provisioned, averaged over all the links in the network and 100 evaluation runs, for the RL agent solution and heuristics on NSFNET. The RL agent distributes channels more uniformly than the heuristics, with no clear bias towards channel ID.

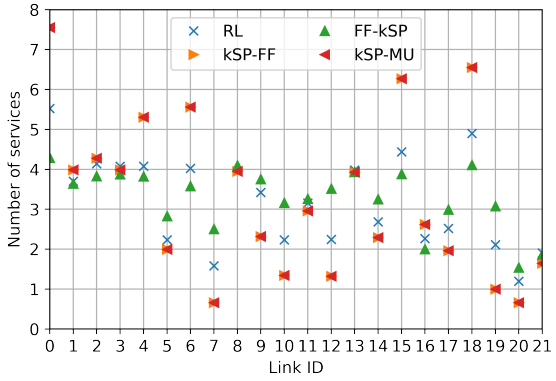


Fig. 15. Link service distribution on NSNFET after 3000 services provisioned, averaged over all channels and 100 evaluation runs, for the RL agent solution and heuristics. The RL agent distribution across links is similar to the heuristics early in the episode.

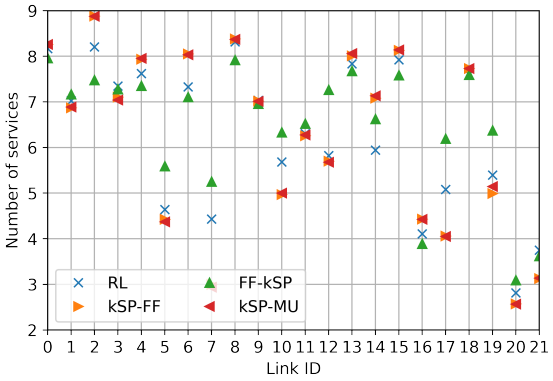


Fig. 16. Link service distribution on NSNFET after 6000 services provisioned, averaged over all channels and 100 evaluation runs, for the RL agent solution and heuristics. The RL agent allocates a similar distribution across links to the heuristics.

heuristics consistently across two different traffic matrices with 99.5% statistical significance for the two considered benchmark topologies NFSNET and COST-239. Crucially, we demonstrate that the RWA policy learned by training the agent on uniformly-distributed traffic generalizes well to a realistic non-uniform traffic distribution unseen during training, outperforming the heuristics. This demonstrates that the policy learned by the RL agent has an ability to generalize to a realistic unseen traffic distribution. Finally, we visualize the distribution of services for the trained RL agent and heuristics in order to interpret some of the key characteristics of the learned RWA policy. The agent shows no bias in terms of the channel IDs chosen for each request, distributing services uniformly across a range of channels. Moreover, the agent distributes services across the links in a way that differs from the heuristics, indicating that the agent has learned a policy that is significantly different. Furthermore, we evaluate the computational cost of using the proposed RL method over the heuristics, both in terms of training the RL agent and the run time for performing RWA once trained. As the run time is similar to the heuristics and the RL training can be performed offline, the RL model shows strong potential for applicability to real world systems.

Future work will concentrate on extending the simulations to a range of network topologies to investigate the scalability and real world applicability of the proposed approach. Moreover, we will extend comparison to a greater range of baselines, including upper bounding global RWA solutions such as integer linear programs. For the two topologies considered, we have observed topology sensitivity of the approach in terms of the magnitude of improvement relative to the heuristics. To this end, we will investigate the potential of incorporating graph neural networks to represent the PPO's policy network to achieve a topology invariant RL solution. Moreover, a rigorous analysis of other RL design parameters, such as the reward structure and observation space, also forms part of the planned future work. Additionally, exploration of the effectiveness of the RWA solution for brown-field RWA scenarios is planned as future work. Also, in future the proposed scheme will be applied to flex-grid elastic optical networks. Due to the potential for large action spaces to arise in these problems, invalid action masking is expected to yield a substantial benefit over standard *tabula rasa* formulations of RL for this problem.

ACKNOWLEDGMENTS

Authors thank the EPSRC through TRANSNET (EP/R035342/1) and the IPES CDT (EP/L015455/1) and BT and Huawei for funding. For the purpose of open access, the author has applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising from this submission.

REFERENCES

1. I. Chlamtac, A. Ganz, and G. Karmi, "Lightpath communications: an approach to high bandwidth optical WAN's," IEEE Trans. Commun. **40**, 1171–1182 (1992).
2. V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," in *NIPS Deep Learning Workshop,* (2013).
3. J. Degrave, F. Felici, J. Buchli, M. Neunert, B. Tracey, F. Carpanese, T. Ewalds, R. Hafner, A. Abdolmaleki, D. de Las Casas *et al.*, "Magnetic control of tokamak plasmas through deep reinforcement learning," Nature. **602**, 414–419 (2022).
4. I. Bello, H. Pham, Q. Le, M. Norouzi, and S. Bengio, "Neural combinatorial optimization with reinforcement learning," in *International Conference on Learning Representations (ICLR),* (2017).
5. K. Li, T. Zhang, and R. Wang, "Deep reinforcement learning for multiobjective optimization," IEEE Trans. Cybern. (2020).
6. R. Shiraki, Y. Mori, H. Hasegawa, and K. Sato, "Dynamic control of transparent optical networks with adaptive state-value assessment enabled by reinforcement learning," in *Proc. Int. Conf. Transparent Opt. Netw. (ICTON),* (IEEE, 2019), pp. 1–4.
7. N. D. Cicco, E. F. Mercan, O. Karandin, O. Ayoub, S. Troia, F. Musumeci, and M. Tornatore, "On deep reinforcement learning for static routing and wavelength assigment," IEEE J. Sel. Top. Quantum Electron. (2022).
8. J. Suárez-Varela, A. Mestres, J. Yu, L. Kuang, H. Feng, P. Barlet-Ros, and A. Cabellos-Aparicio, "Routing based on deep reinforcement learning in optical transport networks," in *Opt. Fiber Commun. Conf. Exhib. (OFC),* (2019), pp. 1–3.
9. C. Xu, W. Zhuang, and H. Zhang, "A deep-reinforcement learning approach for SDN routing optimization," in *Proc. International Conference on Computer Science and Application Engineering,* (2020), pp. 1–5.
10. X. Chen, B. Li, R. Proietti, H. Lu, Z. Zhu, and S. Yoo, "DeepRMSA: A deep reinforcement learning framework for routing, modulation and spectrum assignment in elastic optical networks," J. Light. Technol. **37**, 4155–4163 (2019).
11. L. Xiao, C. Shi, L. Wang, X. Chen, Y. Li, and T. Yang, "Leveraging double-agent-based deep reinforcement learning to global optimization of elastic optical networks with enhanced survivability," Opt. Express **27**, 7896–7911 (2019).
12. N. E. D. E. Sheikh, E. Paz, J. Pinto, and A. Beghelli, "Multi-band provisioning in dynamic elastic optical networks: a comparative study of a heuristic and a deep reinforcement learning approach," in *Conf. Opt. Netw. Des. Model. (ONDM),* (2021), pp. 1–3.
13. Z. Chen, J. Zhang, B. Zhang, R. Wang, H. Ma, and Y. Ji, "ADMIRE: Demonstration of collaborative data-driven and model-driven intelligent routing engine for IP/optical cross-layer optimization in X-haul networks," in *Opt. Fiber Commun. Conf. Exhib. (OFC),* (IEEE, 2022), pp. 1–3.
14. C. Natalino and P. Monti, "The Optical RL-Gym: An open-source toolkit for applying reinforcement learning in optical networks," in *Int. Conf. Transparent Opt. Netw. (ICTON),* (2020), p. Mo.C1.1.
15. M. Minsky, "Steps toward artificial intelligence," Proc. IRE **49**, 8–30 (1961).
16. R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction* (MIT press, 2018).
17. B. Jaumard, C. Meyer, and B. Thiongane, "Comparison of ilp formulations for the rwa problem," Opt. Switch. Netw. **4**, 157–172 (2007).
18. H. Zang and J. P. Jue, "A review of routing and wavelength assignment approaches for wavelength-routed optical wdm networks," Opt. Networks Mag. **1**, 47–60 (2000).
19. R. J. Vincent, D. J. Ives, and S. J. Savory, "Scalable capacity estimation for nonlinear elastic all-optical core networks," J. Light. Technol. **37**, 5380–5391 (2019).
20. C. Xu, W. Zhuang, and H. Zhang, "A deep-reinforcement learning approach for SDN routing optimization," in *International Conference on Computer Science and Application Engineering,* (Association for Computing Machinery, New York, NY, USA, 2020), CSAE 2020.
21. K. Rusek, J. Suárez-Varela, P. Almasan, P. Barlet-Ros, and A. Cabellos-Aparicio, "RouteNet: Leveraging graph neural networks for network modeling and optimization in sdn," IEEE J. Sel. Areas Commun. **38**, 2260–2270 (2020).
22. N. Luong, D. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, and D. Kim, "Applications of deep reinforcement learning in communications and networking: A survey," IEEE Commun. Surv. Tutor. **21**, 3133–3174 (2019).
23. Q. Zhuge, X. Zeng, H. Lun, M. Cai, X. Liu, L. Yi, and W. Hu, "Application of machine learning in fiber nonlinearity modeling and monitoring for elastic optical networks," J. Light. Technol. **37**, 3055–3063 (2019).
24. X. Jiang, D. Wang, Q. Fan, M. Zhang, C. Lu, and A. Lau, "Solving the nonlinear Schrödinger equation in optical fibers using physics-informed neural network," in *Opt. Fiber Commun. Conf. Exhib. (OFC),* (IEEE, 2021), pp. 1–3.
25. J. W. Nevin, F. J. Vaquero-Caballero, D. J. Ives, and S. J. Savory, "Physics-informed Gaussian process regression for optical fiber communication systems," J. Light. Technol. **39**, 6833–6844 (2021).
26. C. Häger and H. D. Pfister, "Nonlinear interference mitigation via deep neural networks," in *Proc. Opt. Fiber Commun. Conf. (OFC),* (IEEE, 2018), pp. 1–3.
27. P. Poggiolini, "The GN model of non-linear propagation in uncompensated coherent optical systems," J. Light. Technol. **30**, 3857–3879 (2012).
28. N. A. Shevchenko, S. Nallaperuma, and S. J. Savory, "Maximizing the information throughput of ultra-wideband fiber-optic communication systems," Opt. Express **30**, 19320–19331 (2022).
29. J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," CoRR **abs/1707.06347** (2017).
30. S. Huang and S. Ontañón, "A closer look at invalid action masking in policy gradient algorithms," CoRR **abs/2006.14171** (2020).
31. Z. Shabka and G. Zervas, "Resource allocation in disaggregated data centre systems with reinforcement learning," arXiv e-prints pp. arXiv–2106 (2021).
32. S. Avci and E. Ayanoglu, "Network coding-based link failure recovery over large arbitrary networks," in *2013 IEEE Global Communications Conference (GLOBECOM),* (IEEE, 2013), pp. 1519–1525.
33. A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-Baselines3: Reliable reinforcement learning implementations," J. Mach. Learn. Res. **22**, 1–8 (2021).
34. D. J. Ives, P. Bayvel, and S. J. Savory, "Routing, modulation, spectrum and launch power assignment to maximize the traffic throughput of a nonlinear optical mesh network," Photonic Netw. Commun. (2015).
35. D. G. Pereira, A. Afonso, and F. M. Medeiros, "Overview of Friedman's test and post-hoc analysis," Commun. Stat. - Simul. Comput. **44**, 2636–2653 (2015).
36. "United States Census Bureau Decennial census P.L. 94-171 redistricting data," https://www.census.gov/programs-surveys/decennial-census/about/rdo/summary-files.html. Accessed: 2021-03-29.