Multi-Agent and Cooperative Deep Reinforcement Learning for Scalable Network Automation in Multi-Domain SD-EONs

Baojia Li, Ruyun Zhang, Xiaojian Tian, and Zuqing Zhu, *Senior Member, IEEE*

***Abstract*—The service provisioning in multi-domain software-defined elastic optical networks (SD-EONs) is an interesting but difficult problem to tackle, because the basic problem of lightpath provisioning, *i.e.*, the routing and spectrum assignment (RSA), is $\mathcal{NP}$-hard, and each domain is owned and operated by a different carrier. Therefore, even though numerous RSA heuristics have been proposed, there does not exist a universal winner that can always achieve the lowest blocking probability in all the scenarios of a multi-domain SD-EON. This motivates us to revisit the inter-domain provisioning problem in this paper by leveraging deep reinforcement learning (DRL). Specifically, we propose DeepCoop, which is an inter-domain service framework that uses multiple cooperative DRL agents to achieve scalable network automation in a multi-domain SD-EON. DeepCoop employs a DRL agent in each domain to optimize intra-domain service provisioning, while a domain-level path computation element (PCE) is introduced to obtain the sequence of the domains to go through for each lightpath request. By sharing a restricted amount of information among each other, the DRL agents can make their decisions distributedly. To ensure scalability and universality, we design the action space of each DRL agent based on well-known RSA heuristics, and architect the agents based on the soft actor-critic (SAC) scenario. We run extensive simulations to evaluate DeepCoop, and the results show that DeepCoop can adapt to the dynamic environment in a multi-domain SD-EON to always select the best RSA heuristic for minimizing blocking probability, and it outperforms the existing algorithms on inter-domain provisioning in various scenarios. Moreover, we verify that the distributed training implemented in DeepCoop ensures its universality and scalability (*i.e.*, its training and operation do not depend on the topology of the SD-EON).**

***Index Terms*—Multi-agent system, Deep reinforcement learning (DRL), Software-defined networking (SDN), Elastic optical networks (EONs), Multi-domain, Network automation.**

I. Introduction

Backbone networks are recently undergoing dramatic changes to adapt to the rising of new network paradigms (*e.g.*, cloud computing, virtualization, 5G, and Internet-of-things (IoT)) [1–6]. This stimulated intensive interests on developing highly efficient, flexible and scalable optical networking technologies. Hence, flexible-grid elastic optical networks (EONs), which possess an agile optical layer and thus can manage optical spectra more flexibly and spectrum-efficiently than traditional fixed-grid wavelength-division multiplexing (WDM) networks [7–9], have been recognized as a promising backbone infrastructure for future Internet.

Meanwhile, software-defined networking (SDN) [10, 11] can be leveraged to realize software-defined EONs (SD-EONs) [12–14]. Note that, for optical networks, the control and data planes were already separated before the introduction of SDN, as defined in the generalized multi-protocol label switching (GMPLS) architecture [15]. Therefore, the actual innovation of SDN on optical networks is the introduction of a centralized controller for network control and management (NC&M) and centralized signaling protocols (*e.g.*, OpenFlow [10]), with which the advantages of EONs on spectrum management and network programmability can be further explored [16].

As a backbone network can span over a relatively large geographical area and/or include network elements produced by multiple vendors, it is usually operated by more than one carriers, each of which manages an autonomous domain. Hence, we should extend the research on SD-EONs to address the multi-domain scenario [15, 17, 18]. Specifically, for the multi-domain scenario, we assume that each domain is owned and operated by a different carrier, and thus optical-to-electrical-to-optical (O/E/O) conversions are applied on both sides of each inter-domain link to protect domain autonomy and privacy [17]. Here, one of the most challenging problems is how to serve inter-domain lightpath requests cost-effectively and time-efficiently, in consideration of the autonomy of each domain and scalability issues. Meanwhile, we need to point out that in certain multi-domain SD-EONs, inter-domain lightpaths can also be set up all-optically end-to-end [19].

The rationale behind this is three-fold. First of all, the fundamental problem of service provisioning in EONs, *i.e.*, the routing and spectrum assignment (RSA), is $\mathcal{NP}$-hard even for its single-domain version [20]. Secondly, numerous heuristics have been designed to solve RSA in various EON scenarios [21], and thus it will be difficult to choose a proper heuristic even if we do not require a guaranteed performance gap to the optimal solution. Lastly but most importantly, to ensure the autonomy of each domain, a domain manager (DM) will not disclose detailed intra-domain information to its peer DMs or the domain-level path computation element (PCE) [22], and thus it would be challenging to coordinate DMs and the domain-level PCE for high-quality inter-domain service provisioning (*i.e.*, balancing the tradeoff between the optimality of service provisioning and the autonomy of domains).

Recently, deep reinforcement learning (DRL) has been widely admitted as a powerful tool that can make timely and

B. Li, X. Tian and Z. Zhu are with the School of Information Science and Technology, University of Science and Technology of China, Hefei, Anhui 230027, P. R. China (email: zqzhu@ieee.org).

R. Zhang is with the Research Center for Industrial Internet, Zhejiang Lab, Hangzhou, Zhejiang 311121, P. R. China.

Manuscript received on October 25, 2020.

smart decisions to solve complex optimizations in dynamic environments [23]. Specifically, DRL leverages one or more agents, each of which consists of and optimizes deep neural networks (DNNs), to interact with a dynamic environment and find the strategy for making the best decision [24]. This feature opens up a lot of new opportunities to address the complex optimizations in NC&M. Therefore, the symbiosis of SDN and DRL has been considered as the most promising method to realize network automation, and various DRL models have been proposed to explore its benefits [25–27].

Previously, people leveraged DRL to propose DeepRMSA [28] to solve RSA in a single-domain EON. Their simulation results suggested that after being trained with $5,000,000$ requests, DeepRMSA could outperform two well-known heuristics (*i.e.*, the shortest-path routing and first-fit (SP-FF) and K-shortest-path routing and first-fit (KSP-FF)), and compared with the better benchmark (KSP-FF), it could reduce the blocking probability by 20.3%. Meanwhile, DeepRMSA was also considered in a multi-domain SD-EON, to address inter-domain provisioning [29]. However, DeepRMSA only has limited scalability and universality[1], because it chooses the actual RSA scheme (*i.e.*, the path and frequency slots (FS') on it) for each lightpath. Specifically, the size of DeepRMSA's action space will increase dramatically, if the operator wants to improve its performance and thus considers more routing paths and more FS blocks on each path for a lightpath. Furthermore, since the definition of the actions in DeepRMSA is tightly related to the parameters of an EON (*e.g.*, the topology and number of FS' on each link), the DRL model that has been trained in the EON would become inapplicable in another one.

The aforementioned drawbacks motivated us to propose DeepCoop in [30], which utilizes multiple cooperative DRL agents to achieve scalable network automation in a multi-domain SD-EON. For each lightpath request, DeepCoop first uses a domain-level PCE to obtain the sequence of the domains to go through, and then relies on the DRL agents assigned to the related domains to 1) obtain intra-domain RSA schemes and 2) select proper inter-domain links to concatenate the intra-domain path segments. By only sharing a restricted amount of information among each other, the DRL agents can make their decisions distributedly. To ensure scalability and universality, we designed the action space of each DRL agent based on well-known RSA heuristics. In other words, instead of selecting the RSA scheme directly, a DRL agent in DeepCoop chooses a proper RSA heuristic from its algorithm pool based on the current network status, and then uses the heuristic to calculate the RSA scheme for a lightpath request. Hence, the action space becomes significantly smaller, and it is independent of the parameters of an EON.

Although our preliminary study in [30] has already confirmed the scalability and universality of DeepCoop and verified that it can outperform existing benchmarks, its performance can still be improved. Hence, this paper expands it to make the problem-solving more comprehensive, with the following improvements. Firstly, we re-architect its DRL agents based on the soft actor-critic (SAC) scenario, which can achieve a better tradeoff between exploration and exploitation [31] than the advantage actor critic (A2C) considered in [30]. This effectively prevents DeepCoop from being trapped by local optima, so that it can perform better in a multi-domain SD-EON with many domains. Meanwhile, by sharing limited state information and obtaining their rewards cooperatively, the DRL agents can converge faster in distributed online training.

Secondly, we redesign the action and state spaces of each DRL agent to make DeepCoop more universal. Note that, for a lightpath request, the DRL agent assigned to each related domain needs to 1) select a RSA heuristic, and 2) choose an inter-domain link to go to the next domain. Our design in [30] architected the action space based on the second task, which made it domain-specific since inter-domain links can be different between different domain pairs. Hence, this work redesigns the action space to fix its dimension over domains. Similarly, to unify the state spaces of DRL agents, we classify state information as intra-domain and inter-domain features, and represent them with feature vectors. Thirdly, we improve the algorithm used by the domain-level PCE for calculating the domain sequence of each lightpath request. Hence, it can work better with the DRL agents to reduce the blocking probability. Finally, we conduct extensive simulations with a multi-domain SD-EON whose topology is much larger than the one used in [30] to evaluate our proposal and verify its effectiveness.

The rest of paper is organized as follows. Section II provides a brief survey on the related work. We present the architecture of DeepCoop and its operation principle in Section III. The detailed design of the multi-agent and cooperative DRL model for DeepCoop is introduced in Section IV. We evaluate the performance of our proposal with numerical simulations in the Section V. Finally, Section VI summaries the paper.

II. Related Work

To facilitate service provisioning, the problem of RSA and its variants have been studied intensively since the inception of EONs. Previous investigations have covered the RSA algorithms for almost all types of communications, including unicast [32–34], multicast [35–37], anycast [38], *etc*. For a comprehensive tutorial on RSA algorithms, one is recommended to refer to [21]. Although for a given EON, the optimal RSA scheme of one lightpath can be obtained time-efficiently with the breadth-first search, optimizing the RSA schemes for multiple lightpaths jointly is $\mathcal{NP}$-hard [20]. Most of the existing RSA algorithms are time-efficient heuristics, which cannot provide performance guarantee and might only perform well for certain specific scenarios. Hence, choosing the right RSA algorithm will be a hassle, especially when the EON has a time-variant environment. This is because in dynamic EONs, lightpath requests can be blocked for various reasons, and thus a heuristic whose objective is deterministic cannot ensure the smallest blocking probability all the time [39].

By implementing RSA algorithms in the control plane, people have conducted experiments to evaluate the performance of single-domain SD-EONs on cost-effectiveness, programmability and resiliency [40–43]. The control plane architecture of

[1] Here, the universality means that a DRL-based approach's design is generic to the topology and resource configuration of a multi-domain SD-EON.

multi-domain SD-EONs has been considered in [15, 17, 44–46]. The proposals in [15, 17, 44] let the SDN controllers of domains (*i.e.*, the DMs) cooperate in a peer-to-peer manner, and utilized the flat control plane architecture for inter-domain service provisioning. As there is no domain-level PCE in the flat architecture, it might take relatively long time to provision each inter-domain lightpath in a multi-domain SD-EON.

In order to address this scalability issue, the hierarchical architecture, which uses a domain-level PCE (*i.e.*, a broker) to coordinate the DMs for inter-domain provisioning, was proposed in [45]. Specifically, for each lightpath request, the domain-level PCE first determines the sequence of the domains to go through, based on the intra-domain virtualized topologies (ID-VTs) provided by the DMs and the status of inter-domain links, and then the DM of each selected domain calculates the RSA scheme for the lightpath segment within its domain. Hence, inter-domain provisioning can be realized without violating the autonomy of each domain. To further protect the domain autonomy, a market-driven multi-broker architecture was designed in [46] to introduce multiple domain-level PCEs for avoiding a single one playing the role of monopoly.

The studies in [17, 44] considered the algorithms for realizing inter-domain provisioning in the multi-domain SD-EONs that use the flat control plane architecture. Considering the hierarchical architecture that consists of multiple domain-level PCEs, we leveraged game theory to tackle the inter-domain provisioning in it in [18, 47]. One interesting observation in [47] is that by mixing the usages of two RSA heuristics adaptively, one can achieve lower blocking probability than using any one of the heuristics constantly. This actually motivates us to design the action space of the DRL model used in this work based on a few well-known RSA heuristics.

Previously, people proposed a game theoretic approach to deal with the dynamic spectrum management in multi-domain wireless networks in [48]. Meanwhile, leveraging the symbiosis of SD-EON and DRL to achieve agile lightpath provisioning has just started to attract research interest since recently [28, 29, 49, 50]. However, these existing approaches designed the action spaces of their DRL models based on the actual RSA schemes for lightpaths, which, as we have explained in the previous section, leads to scalability and universality issues. Therefore, although the study in [29] also tried to utilize multi-agent DRL to solve inter-domain lightpath provisioning, the restrictions on scalability and universality still exist. Moreover, as the proposal in [29] let the DRL agents compete but not cooperate with each other, the operation complexity would actually increase with the number of agents. Multi-agent DRL was also included in the control plane of a multi-domain SD-EON in [51], but it was used for quality-of-transmission (QoT) estimation. Hence, to our best of knowledge, this is the first work that can utilize multiple cooperative DRL agents to realize scalable network automation for the inter-domain service provisioning in multi-domain SD-EONs.

III. Proposed Scalable Inter-domain Provisioning Framework

In this section, we describe the network architecture of DeepCoop and explain the inter-domain provisioning with it.

A. Architecture of DeepCoop

We design DeepCoop to tackle inter-domain lightpath provisioning in a dynamic multi-domain SD-EON, which is a relatively complex problem. Specifically, it involves two subproblems, 1) finding a domain-level path for each inter-domain lightpath (*i.e.*, a sequence of domains from the source to the destination of the lightpath), and 2) calculating a feasible RSA scheme in each domain on the domain-level path.

The overall system architecture of DeepCoop is shown in Fig. 1 [30], which utilizes the hierarchical control plane for inter-domain service provisioning. Specifically, in the multi-domain SD-EON, each domain has a domain manager (DM), which is essentially the controller of all the data plane elements in its domain, and the domain-level PCE is introduced to get the global information about the domains by merging the intra-domain status from the DMs, and coordinate the DMs to set up inter-domain lightpaths accordingly. Here, each DM reports its intra-domain status by abstracting and submitting an intra-domain virtualized topology (ID-VT), which is a simplified topology that only contains aggregated information about the nodes and links in the domain (*i.e.*, the domain's border nodes interconnected with a fully-meshed set of virtual links), for protecting the autonomy and privacy of each domain [47].

To provision an inter-domain lightpath request, the domain-level PCE first collects the information about the request and ID-VTs from the DMs, and then calculates the domain-level routing path for the request. Note that, there are generally two approaches for the domain-level PCE to calculate a domain-level routing path: 1) selecting only the domain sequence (*i.e.*, the sequence of the domains to go through from the lightpath's source to its destination), and 2) selecting the domain sequence together with related border nodes [52]. We design the domain-level PCE to use the first approach because it leaves more space for each DM to optimize intra-domain RSA with DRL.

Each DM includes a DRL agent and an SDN controller. The controller sends the state of the domain and the information about pending lightpath requests (if the domain is the source domain) to the DRL agent, and establishes intra-domain lightpath segments according to the returned provisioning schemes. Meanwhile, the controller also collects the new state of the domain after setting up an intra-domain lightpath segment, and feeds it back to the DRL agent for reward calculation. During training, the DRL agent learns how to analyze the current state of the domain to 1) select a proper RSA heuristic from its algorithm pool to compute an intra-domain lightpath segment, and 2) choose the best inter-domain link for the lightpath segment to connect to its next domain. Meanwhile, the DRL agents share a restricted amount of intra-domain information among each other and calculate rewards collaboratively to improve their performance on service provisioning.

B. Operational Principle of DeepCoop

We model a multi-domain SD-EON with N domains as $\mathbb{G} = \{G^i(V^i, E^i), \forall i \in [1, N],\ \tilde{E}\}$. Here, $G^i(V^i, E^i)$ denotes the intra-domain topology of *Domain* i, where V^i and E^i are the sets of nodes and links in the domain, and $\tilde{E}$ is the set of inter-domain links. Each intra-domain links $e \in E^i$ in

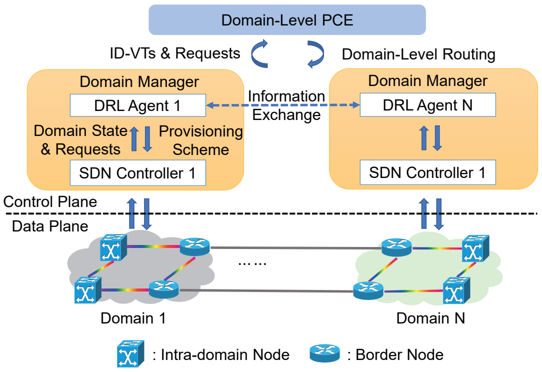


Fig. 1. Network architecture of DeepCoop.

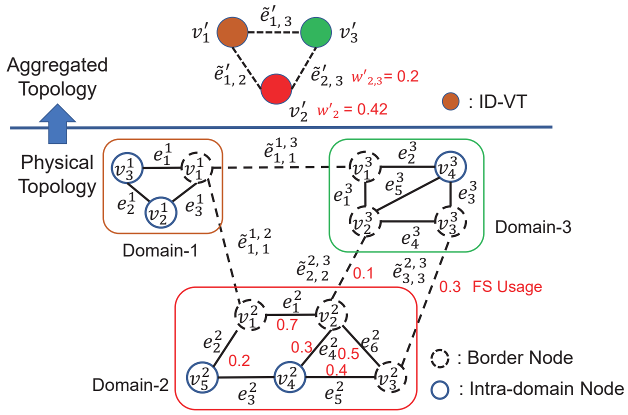


Fig. 2. Example on constructing an aggregated topology.

Domain i can accommodate F^i FS', while each inter-domain link $\tilde{e} \in \tilde{E}$ can support $\tilde{F}$ FS'. As we assume that there are O/E/O conversions on both sides of each inter-domain link, the spectrum continuity constraint only needs to be considered in intra-domain RSA calculation. An inter-domain link between *Domains* i and j can also be denoted as $\tilde{e}^{i,j}_{u,v}$, where $u \in V^i$ and $v \in V^j$ are the border nodes at its two ends.

Note that, the physical topology $\mathbb{G}$ will not be available to the domain-level PCE, and it can only merge the ID-VTs from the DMs to get an aggregated topology. *Algorithm* 1 explains the procedure for obtaining the aggregated topology. *Lines* 2-14 are for the DMs to abstract and report the ID-VTs regarding their domains. Specifically, for each *Domain* i, we first find all the intra-domain links, each of which uses at least one of the border nodes as an end-node, and stores the average FS usage on them in w'_i (*Lines* 4-5). Then, in the *Lines* 6-11, we check all the inter-domain links between *Domain* i and each of its adjacent domains (*e.g.*, *Domain* j), store the average FS usage on them in $w'_{i,j}$, aggregate the inter-domain links as an aggregated link $\tilde{e}'_{i,j}$ between *Domains* i and j, and assign the weight of the aggregated link as $w'_{i,j}$. Next, we abstract the topology of *Domain* i (G^i) as an aggregated node and assign its weight as w'_i, and form an ID-VT with the aggregated node and all the aggregated links that terminate at it to report to the domain-level PCE (*Lines* 12-13). Finally, the domain-level PCE merges all the received ID-VTs to obtain an aggregated topology to represent the multi-domain SD-EON (*Line* 16), and for each inter-domain lightpath, it calculates the least-weighted path P^η in the aggregated topology from the lightpath's source domain to destination domain (*Line* 17). The path P^η is just the domain-level routing path of the lightpath.

Fig. 2 shows an example on how to build an aggregated topology, where the physical topology $\mathbb{G}$ consists of three domains and four inter-domain links. The domains are abstracted into three nodes in the aggregated topology, while the inter-domain links among them are also aggregated correspondingly. For instance, in the aggregated topology, *Domain* 2 is abstracted as virtual node v'_2 and the two inter-domain links between it and *Domain* 3 (*i.e.*, $\tilde{e}^{2,3}_{2,2}$ and $\tilde{e}^{2,3}_{3,3}$), are aggregated as the virtual link $\tilde{e}'_{2,3}$ that connects virtual nodes v'_2 and v'_3. In *Domain* 2, all the intra-domain links that directly connect to the border nodes are in set $E'_2 = \{e^2_1, e^2_2, e^2_4, e^2_5, e^2_6\}$, and the FS usages on them are $\{0.7, 0.2, 0.3, 0.4, 0.5\}$, respectively, which are marked aside the links in Fig. 2. Hence, the weight of virtual node v'_2 should be $w'_2 = 0.42$ (*i.e.*, the average FS usage of all the links in E'_2), and the weight of virtual link $\tilde{e}'_{2,3}$ should be the average FS usage of $\tilde{e}^{2,3}_{2,2}$ and $\tilde{e}^{2,3}_{3,3}$ (*i.e.*, $w'_{2,3}0.2$).

Algorithm 1: Construction of Aggregated Topology

```
1  Procedure of DMs:
2  for each Domain $i \in [1, N]$ do
3      $E' = \emptyset$;
4      find all the intra-domain links that directly connect to
       border nodes in Domain $i$, and store them in $E'_i$;
5      calculate average FS usage on all the links in $E'_i$ and
       store the value in $w'_i$;
6      for each neighbor Domain $j$ of Domain $i$ do
7          $E' = \emptyset$;
8          find all the inter-domain links that are between
           Domains $i$ and $j$, and store them in $E'_{i,j}$;
9          calculate average FS usage on all the links in
           $E'_{i,j}$ and store the value in $w'_{i,j}$;
10         aggregate all the links in $E'$ as an aggregated
           link $\tilde{e}'_{i,j}$ and assign a weight $w'_{i,j}$ to it;
11     end
12     abstract $G^i(V^i, E^i)$ as an aggregated node and assign
       a weight $w'_i$ to it;
13     connect all aggregated links to the aggregated node
       to form an ID-VT to report to domain-level PCE;
14 end
15 Procedure of Domain-level PCE:
16 collect all the ID-VTs from DMs and merge them into an
   aggregated topology;
17 apply the Dijkstra algorithm to the aggregated topology
   to find the least-weighted path $P^\eta$;
```

With the domain-level routing path P^η, the domain-level PCE can coordinate the related DMs to set up the inter-domain lightpath end-to-end. Specifically, each related DM leverages its DRL agent to determine both the RSA scheme of the lightpath segment in its domain and the inter-domain link to go to the next domain. Note that, instead of making their

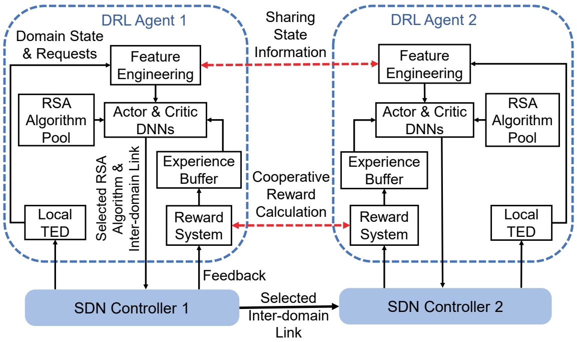


Fig. 3. Cooperation between two DRL agents for inter-domain provisioning.

decisions independently, the DRL agents actually cooperate with each other to improve the performance of inter-domain service provisioning. We use Fig. 3 to briefly explain the cooperation among the DRL agents, while their detailed design and operation principle will be discussed in the next section. Here, we assume that the domain-level path needs to route a lightpath from *Domain* 1 to *Domain* 2. First of all, *DRL Agent* 1 gets the current state of its domain and also fetches the state parameters of *Domain* 2 from *DRL Agent* 2. Then, it selects a proper RSA algorithm and an inter-domain link to go to *Domain* 2 based the state information. Next, the RSA scheme of the lightpath segment in *Domain* 1 is calculated by the selected RSA algorithm. Note that, as the lightpath will experience O/E/O conversion in the related border nodes, we can simply determine its spectrum assignment on the inter-domain link with the first-fit approach. Finally, with the intra-domain RSA scheme and the inter-domain link, the ingress node in *Domain* 2 can be determined.

DRL Agent 2 uses the same procedure to obtain the RSA scheme of the lightpath segment in its domain, but since *Domain* 2 is the destination domain, it does not need to collect the state parameters of the next domain. When the end-to-end RSA scheme of the inter-domain lightpath has been determined, the related DMs instruct their SDN controllers to establish the lightpath and collect the domain states after the service provisioning. Hence, the reward systems of the DRL agents can calculate the rewards of their selected actions collaboratively, for evaluating the whole inter-domain RSA scheme better. Specifically, each DRL agent on the domain-level path forwards its reward to the DRL agent of its previous domain. For instance, in Fig. 3, *DRL Agent* 2 first calculates the reward of its selected action based on the new state in *Domain* 2, and then sends the reward to the reward system in *DRL Agent* 1. Next, *DRL Agent* 1 calculates its own reward based on the reward from *DRL Agent* 2 and the new state in *Domain* 1. After obtaining the reward, each DRL agent stores the reward and its corresponding action and state in its experience buffer as a training sample, which will be leveraged to update its DNNs in the online training.

IV. Multi-agent and Cooperative DRL for Inter-domain Service Provisioning

In this section, we elaborate on the multi-agent DRL model used by DeepCoop for inter-domain service provisioning.

A. Background of Multi-agent DRL

The principle of DRL is about making one or more intelligent agents learn on how to act to maximize the reward by interacting with a dynamic environment constantly. The learning process can be modeled as a Markov decision process [24], which is defined by a tuple $\{\mathbf{S}, A, R, \mathbf{P}\}$. Here, $\mathbf{S}$ represents the state space (*i.e.*, the set of all the states of the environment), A denotes the action space (*i.e.*, the set of all the actions that the agent(s) can take), R is the reward function that can be used to calculate the reward (*i.e.*, the Q-value) obtained by an agent after it applying an action $a \in A$ in respond to a state $S \in \mathbf{S}$, and $\mathbf{P}$ is the matrix that describes the transition probabilities of the states. The objective of a DRL is to find the optimal policy π^*, which can map each state $S \in \mathbf{S}$ to a proper action $a \in A$ such that the reward defined by R can be maximized. Note that, a reward usually contains two parts, which are the immediate reward r and state value δ, and in practice, it can be approximated by defining R as

$$R(a_t, S_t) = \sum_t \gamma^t \cdot r_t, \tag{1}$$

where t is the time instant, γ is the discount factor, and r_t is the immediate reward at time t.

Multi-agent DRL makes several DRL agents work on cooperative task(s) to achieve global optimality, and thus we assign a DRL agent to each DM and leverage them to realize high-performance inter-domain service provisioning. Note that, in a multi-agent DRL model, the DRL agent can cooperate in two ways. The first one is that the agents will not communicate with each other, and their cooperation is coordinated with a central critic neural network (C-NN). Specifically, the central C-NN can observe the operations of all the DRL agents, estimate their Q-values, and coordinate their cooperative actions accordingly [53]. Nevertheless, for the service provisioning in a multi-domain SD-EON, introducing a central C-NN would limit the scalability of the NC&M and damage the autonomy of the domains. Moreover, the provisioning of an inter-domain lightpath might not involve all the DMs, and thus using the central C-NN to evaluate the actions of all the DRL agents constantly is not only unnecessary but also misleading. Therefore, we turn to the second way that lets the agents communicate with each other for enabling cooperation [54].

B. Modeling Inter-domain Provisioning with Multi-agent DRL

As we have explained in the previous section, the DRL agent in each DM needs to select both the RSA algorithm to compute the lightpath segment in its domain and the inter-domain link to go to the next domain. Hence, its action affects not only the service provisioning in its own domain but also that in the next domain. To this end, we design the state of each DRL agent to include the information about the current and next domains, and formulate the reward function to consider the new state of the current domain and the feedback from the next domain. The model of each DRL agent is as follows.

State: At time instant t, the state $\mathbf{S}_t^j$ observed by DRL agent Ψ^j in *Domain* j contains the information about the current and next domains. For the current domain, its state is represented by the status of a few paths, which are from the source node in it to the border nodes that connect to the next domain. Here, the source node is just the source of the inter-domain lightpath if the current domain is the source domain, and it is where the lightpath enters the current domain, otherwise[2]. For each pair of the source node and a feasible border node, we calculate K shortest paths in the current domain, and record three parameters about each path as its state, which are 1) the average size of available FS blocks, 2) the number of available FS blocks, and 3) the start-index of the first available FS block, for serving the lightpath request on the path. Hence, the information about the node pair can be represented by a feature vector that includes all the parameters of the K paths, where the length of the feature vector is $3{\cdot}K$. With the physical topology $\mathbb{G}$, we can get the maximum number of feature vectors for a domain as N_b. Therefore, the information about the current domain can finally be denoted with N_b feature vectors $\{\Phi_i^{\text{intra}}, i \in [1, N_b]\}$, and if the actual number of feature vectors is less than N_b, we fill the remaining ones with zeros.

For the next domain, its state is represented by the status of 1) all the inter-domain links between it and the current domain and 2) a few related paths in it. We record the FS usage on each inter-domain link. Then, as each inter-domain link determines the ingress border node in the next domain, the information about the pair of the ingress border node and a feasible egress border node[3] is also recorded. Specifically, the DRL agent of the next domain calculates K shortest paths between the two border nodes, and reports the average values of 1) the sizes of available FS blocks, 2) the numbers of available FS blocks, and 3) the start-indices of the first available FS blocks on the paths, to the DRL agent of the current domain. Hence, for the next domain, each feature vector (with a length of 4) includes the FS usage on an inter-domain link and the three average values about paths between a related border node pair in it.

For instance, in Fig. 2, if we need to set up a lightpath from v_3^1 to v_4^3 and its domain-level routing path is *Domain* 1 $\rightarrow$ *Domain* 2 $\rightarrow$ *Domain* 3, the DRL agent for *Domain* 1 can get two feature vectors about the next domain (*i.e.*, *Domain* 2). The two vectors store the information about $\{\tilde{e}_{1,1}^{1,2},\ v_1^2 \rightarrow v_2^2\}$ and $\{\tilde{e}_{1,1}^{1,2},\ v_1^2 \rightarrow v_3^2\}$. Here, $v_1^2 \rightarrow v_2^2$ refers to a pair of ingress border node and egress border node, and features of the paths between them are considered for the next domain state. We also fix the number of this type of feature vectors as its maximal value (*i.e.*, N_l), and thus the information about the next domain is denoted with N_l feature vectors $\{\Phi_i^{\text{inter}}, i \in [1, N_l]\}$. Similarly, if the actual number of feature vectors is less than N_l, we append with zero filling. Finally, by combining the feature vectors $\{\Phi_i^{\text{intra}}, i \in [1, N_b]\}$ and $\{\Phi_i^{\text{inter}}, i \in [1, N_l]\}$, we obtain the state of a DRL agent.

Action: At time instant t, the action a_t^j taken by DRL agent Ψ^j in respond to state $\mathbf{S}_t^j$ includes a RSA algorithm to compute the lightpath segment *Domain* j and the inter-domain link for the lightpath go to the next domain. Hence, each action can be represented as a tuple $\{\omega^{\text{RSA}}, \tilde{e}\}$, where the RSA algorithm ω^{RSA} is from an algorithm pool Ω^{RSA}, and $\tilde{e}$ is the selected inter-domain link. In this work, we select four well-known RSA heuristics to build the algorithm pool Ω^{RSA}, which are the K-shortest path and first-fit (KSP-FF), K-shortest path and load-balancing (KSP-LB), K-shortest path and fragmentation-aware (KSP-FA), and fragmentation-aware and congestion-avoidance (FA-CA).

These RSA heuristics are considered because they are time-efficient, and each of them performs well in certain condition [21]. To provision a lightpath, KSP-FF selects the shortest path in hops and uses the first-fit scheme for spectrum assignment (*i.e.*, it helps to reduce the overall spectrum usage), KSP-LB checks K shortest paths and tries to balance the spectrum usages on them (*i.e.*, it can avoid causing bottleneck links especially when the EON is relatively crowded), KSP-FA also considers K shortest paths but tries to induce the least spectrum fragmentation, and FA-CA tries to not only cause the least spectrum fragmentation but also balance the spectrum usages on K shortest paths. In this work, we follow their original designs to implement the RSA heuristics.

Reward: The objective of the inter-domain service provisioning is to minimize the blocking probability of lightpath requests. Hence, as each action a_t^j taken by DRL agent Ψ^j determines not only the algorithm for intra-domain RSA calculation but also the inter-domain link to the next domain, the instant reward r_t^j of DRL agent Ψ^j for the action should relate to whether or not the lightpath request can be successfully provisioned in the current and next domains. Specifically, if the lightpath can be provisioned in a domain, a positive reward is returned by the domain, and the reward is negative, otherwise. Then, the instant reward r_t^j is obtained by summarizing the rewards from the current and next domains.

However, it would be difficult to accurately evaluate the actions, if we only consider whether the lightpaths can be provisioned. Therefore, the positive reward of successful provisioning in the current domain is defined as the sum of 1) the FS availability on the path with the minimum spectrum usage in states S_{t+1}^j and S_t^j, and 2) the ratio of the size of the largest FS block on the lightpath's path candidates in the current domain in S_{t+1}^j to that in S_t^j. This encourages DRL agents to select the actions that can leave more spectra for future requests and thus lead to lower long-term blocking probability. For the positive reward in the next domain, we define it as the difference between the maximum and minimum FS availabilities on the paths between its ingress and egress border nodes in states S_t^j and S_{t+1}^j, to ensure load-balancing in the next domain. The negative rewards of the current and next domains are set as -1. Note that, if the current domain is the destination domain, the reward in the next domain is 0.

C. Design of DRL Agent

We design the structure of each DRL agent based on the soft actor-critic (SAC) scenario [31], which tries to maximize not only the long-term reward in Eq. (1) but also the policy entropy that measures the randomness of policy selection. By doing so, the DRL agent based on SAC is encouraged to explore

[2] Note that, as each lightpath is sequentially served from source to destination by the DMs along the domain-level routing path, each DRL agent should know the source node in its domain when it is invoked.

[3] Here, a feasible egress border node is one node that connects to the domain after the next domain, according to the domain-level routing path.

its action space more thoroughly and reduce the possibility of premature convergence in training. Specifically, the training of DRL agent Ψ^j needs to find the optimal policy π^* as

$$\pi^* = \operatorname*{argmax}_{\pi^j} \mathbf{E}_{\tau\sim\pi}\left\{\sum_{t=0}^{\infty}\gamma^t\left[r_t^j+\alpha\cdot H(\pi^j(A^j|S_t^j))\right]\right\}, \tag{2}$$

where $\mathbf{E}(\cdot)$ calculates the expectation of all the state-action pairs generated by a policy, π^j is the DRL agent's policy that can be parameterized by its actor neural network (A-NN) with parameters θ_a, τ denotes the trajectories induced by policy π^j, and r_t^j is the immediate reward after taking action a_t^j at state S_t^j, $H(\cdot)$ calculates the entropy of policy π^j as

$$H[\pi^j(A^j|S_t^j)] = -\sum_{a^j\in A^j}\log(a^j|S_t^j), \tag{3}$$

$\gamma\in[0,1]$ is the discount factor for long term reward calculation and α is weighting factor that determines the relative importance of the immediate reward and the policy entropy. As α actually balances the tradeoff between exploration and exploitation, we set it as a learnable parameter and design an training process for it. Because the designs of all the DRL agents are identical, we will not differentiate them and omit the superscript "j" in the following discussions.

To evaluate policy π and improve it in the training, we first need to estimate the Q-value $Q(S_t,a_t)$ (*i.e.*, the long-term reward of taking action a_t at state S_t) and state value $\delta(S_t)$ (*i.e.*, the goodness of state S_t). We define the state value as

$$\delta(S_t) = \mathbf{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t(r_t+\alpha\cdot H(\pi(A|S_t)))\right]. \tag{4}$$

Meanwhile, according to the Bellman equation [24], the Q-value can be approximated as

$$Q(S_t,a_t) = \mathbf{E}_{\tau\sim\pi}\left[r_t+\gamma\cdot\delta(S_{t+1})\right], \tag{5}$$

By combining Eqs. (4) and (5), we obtain the relation between $\delta(S_t)$ and $Q(S_t,a_t)$ as

$$\delta(S_t) = \mathbf{E}_{\tau\sim\pi}\left[Q(S_t,a_t)+\alpha\cdot H(\pi(A|S_t))\right]. \tag{6}$$

Hence, $Q(S_t,a_t)$ can be approximated with $Q(S_{t+1},a_{t+1})$ as

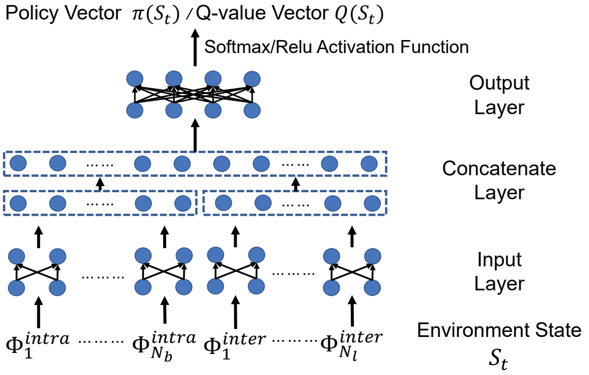
$$Q(S_t,a_t) = \mathbf{E}_{\tau\sim\pi}\left\{r_t+\gamma\cdot[Q(S_{t+1},a_{t+1})+\alpha\cdot H(\pi(A|S_{t+1}))]\right\} \tag{7}$$

As the action space is limited and discrete, the A-NN outputs a policy vector $\pi(S_t)$ (*i.e.*, the distribution of the probabilities to choose each action in A at state S_t), and each $Q(S_t,a_t)$ can be estimated. Hence, we simplify Eq. (6) as

$$\delta(S_t) = \pi(S_t)^T\odot[Q(S_t)-\alpha\cdot\log(\pi(S_t))], \tag{8}$$

where $\pi(S_t)^T$ is the transpose of $\pi(S_t)$, $\odot$ is the inner product for matrices, and $Q(S_t)$ is the vector that includes all the Q-values related to state S_t (*i.e.*, $[Q(S_t,a_t),\ a_t\in A]$).

We design two separate critic neural networks (C-NNs) to parameterize the aforementioned Q-value estimation, and their parameters are $\theta_{c,1}$ and $\theta_{c,2}$, respectively. At each time instant t, they take a state S_t as the input and output two Q-values, *i.e.*, $Q_{c,1}$ and $Q_{c,2}$, respectively. We take the smaller one of $Q_{c,1}$ and $Q_{c,2}$ as the actual Q-value, to avoid the overestimation of state values [55]. Meanwhile, we also design two target C-NNs whose parameters are $\tilde{\theta}_{c,1}$ and $\tilde{\theta}_{c,2}$, respectively. They are used to estimate the Q-value of state S_{t+1}, and their parameters are updated slowly with those of the two original C-NNs, which will be referred to as local C-NNs in the following, as



Policy Vector $\pi(S_t)$ / Q-value Vector $Q(S_t)$
Softmax/Relu Activation Function
Output Layer
Concatenate Layer
Input Layer
Environment State S_t
…… …… …… ……
Φ_1^{intra} ……… $\Phi_{N_b}^{intra}$ Φ_1^{inter} ……… $\Phi_{N_l}^{inter}$

Fig. 4. Structures of A-NN and C-NNs.

$$\begin{aligned}\tilde{\theta}_{c,1} &= (1-\rho)\cdot\tilde{\theta}_{c,1}+\rho\cdot\theta_{c,1},\\ \tilde{\theta}_{c,2} &= (1-\rho)\cdot\tilde{\theta}_{c,2}+\rho\cdot\theta_{c,2},\end{aligned} \tag{9}$$

where $\rho\in(0,1)$ is a constant. We introduce the target C-NNs to stabilize the training process [56]. To enable them to estimate Q-values accurately, we define their loss functions as

$$J_Q(\theta_{c,k}) = \mathbf{E_D}\left\{\frac{1}{2}(Q_{\theta_{c,k}}(S_t)-Q_{\text{targ}})^2\right\}, \quad k\in\{1,2\}, \tag{10}$$

where $\mathbf{E_D}(\cdot)$ means to calculate the expectation over all the training samples in the experience buffer $\mathbf{D}$ (*i.e.*, each training sample is a tuple $\{S_t,a_t,r_t,S_{t+1}\}$), and Q_{targ} is modeled as

$$Q_{\text{targ}} = r_t+\gamma\cdot\pi(S_t)\odot\left\{\min_k\left[Q_{\tilde{\theta}_{c,k}}(S_{t+1})\right]-\alpha\cdot\log\left[\pi(S_{t+1})\right]\right\}. \tag{11}$$

Then, the losses of the two local C-NNs are the mean square error (MSE) between their outputs and Q_{targ}, respectively.

Meanwhile, we define the loss function of the A-NN as

$$J_\pi(\theta_a) = \mathbf{E_D}\left\{\pi(S_t)^T\odot\left\{\min_k\left[Q_{\theta_{c,k}}(S_t)\right]-\alpha\cdot\log\left[\pi_t(S_t)\right]\right\}\right\}, \tag{12}$$

Note that, we set α as a learnable parameter, and thus it should have a loss function too, which is defined as

$$J_\alpha = \mathbf{E_D}\left\{-\alpha\cdot\left\{\log\left[\pi(a_t|S_t)\right]+\tilde{H}\right\}\right\}, \tag{13}$$

where $\tilde{H}$ is a constant scalar that represents the target entropy.

As shown in the Fig. 4, we design the A-NN and C-NNs with similar and universal structures, and make sure that they can be applied to an arbitrary domain in the multi-domain SD-EON. Moreover, as the design of the DRL agent with the A-NN and C-NNs is universal, when a new domain is added, we can initialize its DRL agent with the trained A-NN and C-NNs in an existing domain. This avoids the hassle of training the A-NN and C-NNs from the scratch. The structure in Fig. 4 consists of three layers, *i.e.*, the input, concatenate, and output layers. The input layer uses N_b two-layer and fully-connected neural networks (FC-NNs) to take in the feature vectors of the current domain (*i.e.*, $\{\Phi_i^{\text{intra}}, i\in[1,N_b]\}$), and N_l two-layer FC-NNs to receive the feature vectors of the next domain (*i.e.*, $\{\Phi_i^{\text{inter}}, i\in[1,N_l]\}$). Their activation function is

$$\text{Relu}(x) = \max(0,x), \tag{14}$$

which is widely used in machine learning to avoid the vanishing gradient problem. Then, the concatenate layer organizes the feature vectors $\{\Phi_i^{\text{intra}}, i \in [1, N_b]\}$ and $\{\Phi_i^{\text{inter}}, i \in [1, N_l]\}$ as two long vectors, respectively, from which an FC-NN abstracts features about the current and next domains.

Algorithm 2: Training of Cooperative DRL Agents

```
1  initialize parameters of A-NN and C-NNs for all DRL
   agents $\{\theta^j_a, \theta^j_{c,1}, \theta^j_{c,2},\ \forall j \in [1, N]\}$;
2  $\tilde{\theta}^j_{c,k} = \theta^j_{c,k},\ \forall j \in [1, N], k \in \{1, 2\}$;
3  $\mathbf{D}^j = \emptyset,\ \forall j \in [1, N]$;
4  for each pending lightpath request do
5      release resources occupied by expired requests;
6      calculate domain-level path $P^\eta$ for the lightpath
       request with Algorithm 1;
7      for each Domain $j \in P^\eta$ (source→destination) do
8          get feature vectors to represent state $S^j_t$ ;
9          use A-NN to get action as $a^j_t = \pi^{\theta_a}_t(S^j_t)$;
10         use the RSA heuristic $\omega^{\text{RSA}}$ in $a^j_t$ to calculate
           intra-domain RSA scheme;
11         combine intra-domain RSA with inter-domain
           link $\tilde{e}$ in $a^j_t$ as the overall RSA for Domain $j$;
12         if the RSA scheme can be deployed then
13             get the source node in the next domain;
14             record “provisioned” in reward system ;
15         else
16             record “blocked” in reward system ;
17             break;
18         end
19     end
20     reward systems share provisioning results;
21     update domain states as $\{S^j_t \Rightarrow S^j_{t+1},\ \forall j \in P^\eta\}$;
22     calculate immediate rewards $\{r^j_t,\ \forall j \in P^\eta\}$;
23     insert training sample $\{S^j_t, a^j_t, r^j_t, S^j_{t+1}\}$ into $\mathbf{D}^j$;
24     for each Domain $j \in [1, N]$ do
25         if there are enough training samples in $\mathbf{D}^j$ then
26             for each training step do
27                 randomly select a batch of samples;
28                 get losses with Eqs. (10), (12) and (13);
29                 $\theta^j_{c,k} = \theta^j_{c,k} - \lambda_Q \cdot \nabla_{\theta^j_{c,k}} J_Q,\ \forall k \in \{1, 2\}$;
30                 $\theta^j_a = \theta^j_a - \lambda_\pi \cdot \nabla_{\theta^j_a} J_\pi$;
31                 $\alpha^j = \alpha^j - \lambda_{\alpha^j} \cdot \nabla_{\alpha^j} J_{\alpha^j}$;
32                 $\tilde{\theta}^j_{c,k} = \rho \cdot \theta^j_{c,k} + (1 - \rho) \cdot \tilde{\theta}^j_{c,k}, \forall k \in \{1, 2\}$;
33             end
34         end
35     end
36 end
```

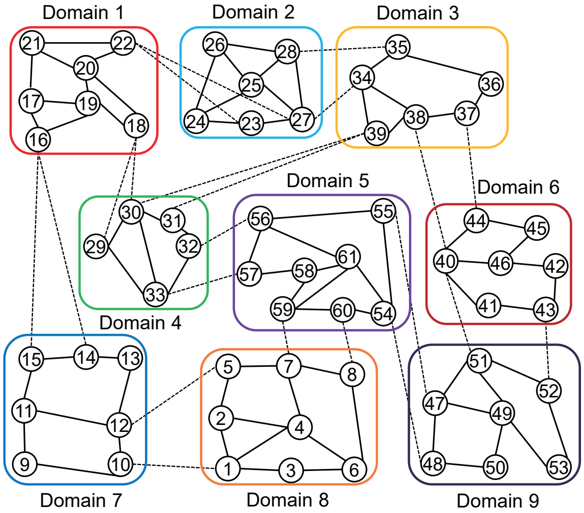


Fig. 5. Topology of 9-domain SD-EON used in simulations.

The designs of the A-NN and C-NN are identical until now, and the only difference lies in their output layers. The output layer of the A-NN uses a two-layer FC-NN with the Softmax activation function to generate the policy vector $\pi(S_t)$ for the current state S_t (*i.e.*, the distribution of probabilities to choose the actions in A). Here, the Softmax activation function is

$$\text{Softmax}(x_i) = \frac{e^{x_i}}{\sum\limits_{j=1}^{M} e^{x_j}}, \tag{15}$$

where x_i is the i-th element in a M-dimension vector X, and it helps to normalize the elements in X. On the other hand, the output layer of a C-NN uses a two-layer FC-NN with the Relu activation function to provide the Q-value vector $Q(S_t)$. Note that, similar to the normal case of DRL design [23], we design and tune the architectures of the A-NN and C-NN and their reward and target Q functions empirically.

D. Training Process of Cooperative Agents

Algorithm 2 explains how to train the cooperative DRL agents for inter-domain provisioning. *Lines* 1-3 are for the initialization, and for each DRL agent, we initialize the parameters of its A-NN, local C-NNs, and target C-NNs, and empty its experience buffer. Then, the for-loop serves pending lightpath requests one-by-one and invokes online training when enough training samples have been accumulated (*Lines* 4-36). Here, *Lines* 5-19 provision an inter-domain lightpath as we have already explained above, while the training samples for the related domains are obtained and inserted in the corresponding experience buffers in *Lines* 20-23. How to leverage online training to update the parameters of neural networks in the DRL agents is explained in *Lines* 24-35. Specifically, for each domain in the multi-domain SD-EON, *Line* 25 checks whether there are sufficient training samples accumulated in the experience buffer. If yes, an online training will be triggered. Then, in each training step, we randomly select a batch of samples from the experience buffer, and use them to calculate the losses of the A-NN, two local C-NNs and weighting factor α (*Lines* 27-28). Next, we obtain their gradients, and use the Adam algorithm [57] with adaptive steps (*i.e.*, λ_Q, λ_π and λ_α) to update the parameters (*Lines* 29-31). Finally, we use the discounted parameters of the local C-NNs to update the parameters of the target C-NNs (*Line* 32).

TABLE I
INPUT/OUTPUT SIZES OF THE A-NN AND C-NNS IN EACH DRL AGENT

Domain	1	2	3	4	5	6	7	8	9
N_b	1	2	2	2	2	2	2	2	2
N_l	2	1	2	2	2	2	2	2	2
Outputs	8	4	8	8	8	8	8	8	8

TABLE II
AVERAGE RUNNING TIME OF OFFLINE TRAINING (SECONDS)

Domain	1	2	3	4	5
Running Time	2,928	2,880	2,823	2,976	3,600
Domain	6	7	8	9	-
Running Time	3,552	2,823	3,246	3,120	-

V. PERFORMANCE EVALUATIONS

In this section, we conduct extensive simulations to evaluate DeepCoop from a few perspectives.

A. Simulation Setup

To demonstrate the scalability of DeepCoop on multi-agent operations, we conduct most of the simulations with a large-scale multi-domain SD-EON that consists of 9 domains with the topology as shown in Fig. 5 [58]. The multi-domain SD-EON contains 61 nodes, 158 intra-domain links and 44 inter-domain links, where each intra-domain and inter-domain link can accommodate 358 FS' and 1074 FS', respectively[4]. We assume that each FS has a bandwidth of 12.5 GHz [20]. The lightpath requests are dynamically generated with the Poisson traffic model, and specifically, we fix the average number of requests arriving in a time-unit as 10 and change the average life time of the requests within $[13, 17]$ time-units, to emulate different traffic loads. Their source and destination nodes are randomly selected from the nodes in the 9 domains, and their bandwidth demands are uniformly distributed within $[2, 9]$ FS'.

The A-NN and C-NNs in a DRL agent adopt the structure that is shown in Fig. 4, and the parameters regarding their input/output sizes are listed in Table I. Here, the values of N_b and N_l determine the input size of the A-NN and C-NNs in each DRL agent. The key hyper-parameters of DeepCoop (*i.e.*, ρ and γ) are empirically set as 0.01 and 0.95. The experience buffer of each DRL agent can store the latest $3,000$ samples. To ensure the statistical accuracy, we run 20 independent simulations and average their results to get each data point. We program DeepCoop with Python, and run the simulations on a high-performance server that equips Intel Xeon E5-2650 CPU, 128 GB RAM, and four GTX 1080ti GPU cards.

To verify the performance of DeepCoop, we consider six benchmarks, four of which are well-known RSA heuristics (*i.e.*, KSP-FF, KSP-LB, KSP-FA, and FA-CA) [21], the fifth one is DeepRMSA [28], and the last one is DeepInd, which is developed here with the similar design of DeepCoop, except for that the DRL agents in it do not share any intra-domain information or calculate reward cooperatively. The benchmarks realize inter-domain provisioning with the operational principle of DeepCoop in Fig. 1. For each benchmark named with a RSA heuristic, all the DMs use the heuristic for intra-domain provisioning, and always choose the feasible inter-domain links that have the minimum spectrum usage to go to next domains. DeepRMSA employs a DRL agent in each domain to decide intra-domain provisioning schemes, and also always chooses the feasible least-used inter-domain links to next domains. Note that, as the action space of DeepRMSA is defined based on intra-domain paths and the FS' on them, DRL agents in different domains cannot share any useful information or cooperate on inter-domain provisioning.

B. Training Performance of DeepCoop

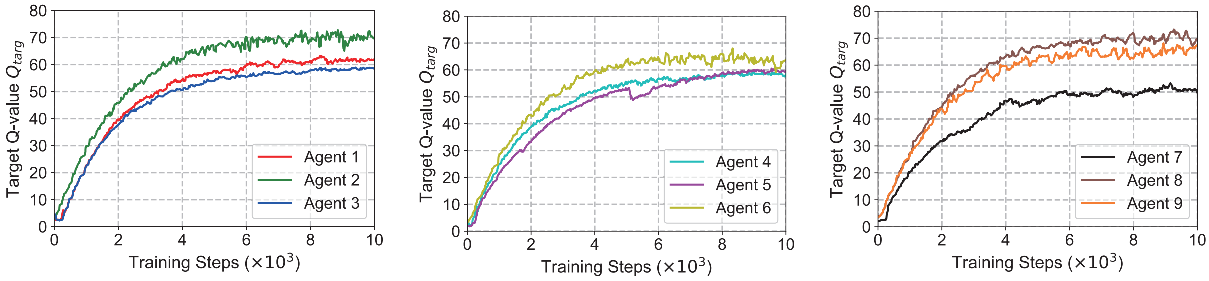
We first evaluate the performance of the training processes of DeepCoop, which train all the DRL agents to maximize their long-term rewards. We hope to point out that the training process of DeepCoop can be manually divided into the offline and online training phases. In the offline training, the learnable parameters of its DRL agents are first initialized randomly, and then optimized to ensure that the DRL agent in each DM becomes suitable for online operation/training. Hence, the offline training should be finished before we can put the DRL agents into operation in the multi-domain SD-EON. Then, in the online operation/training, DeepCoop leverages its DRL agents to provision inter-domain lightpaths, records the provisioning results as new training samples, updates the parameters of the DRL agents with the training samples, and makes itself adapt well to the dynamic network environment of the multi-domain SD-EON. We discuss the running time of the offline training in this subsection, while that of the online operation/training will be analyzed in Section V-D.

As Eq. (11) suggests that the target Q-value (Q_{targ}) can be used to estimate the long-term reward of each DRL agent, we plot how Q_{targ} evolves in the offline training of each agent in Fig. 6. We observe that the values of all the DRL agents' Q_{targ} increase quickly and then converge to stable values within $8,000$ training steps. Specifically, Table II lists the average running time of the offline training of each DRL agent. It can be seen that for all the agents, the average running time of the offline training is always within an hour, and the agents for the domains that sit in the middle of the 9-domain SD-EON and have relatively large numbers of nodes spend the longest time on offline training (*e.g.*, those for *Domains* 5 and 6).

The training performance of DeepCoop can also be verified with the results in Fig. 7, which show the evolution of blocking probability[5] in the training when the number of served lightpath requests increases (*i.e.*, the traffic load is fixed at $1,500$ Erlangs). The blocking probabilities of the RSA heuristics stay

[4] In a multi-domain network, an inter-domain link usually has a larger capacity than an intra-domain one to avoid inter-domain bottlenecks. Here, for each inter-domain link, the number of FS' on it is actually larger than that can be accommodated in the C-band of a fiber. There are two ways to achieve this: 1) using other bands in a fiber, and 2) deploying multiple physical fibers.

[5] Here, we choose blocking probability as the key metrics to evaluate the inter-domain lightpath provisioning algorithms. This is because it relates directly to the revenues of the DMs in a multi-domain SD-EON, *i.e.*, each blocked lightpath request reduces the revenues of the related DMs. Meanwhile, there are clear correlations between it and other metrics (*e.g.*, spectrum usage).



(a) *Agents* 1-3

(b) *Agents* 4-6

(c) *Agents* 7-9

Fig. 6. Evolving of target Q-value Q_{targ} in the offline training.

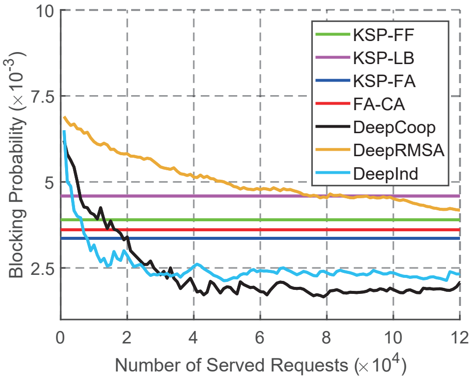


Fig. 7. Evolving of blocking probability (traffic load fixed at $1,500$ Erlangs).

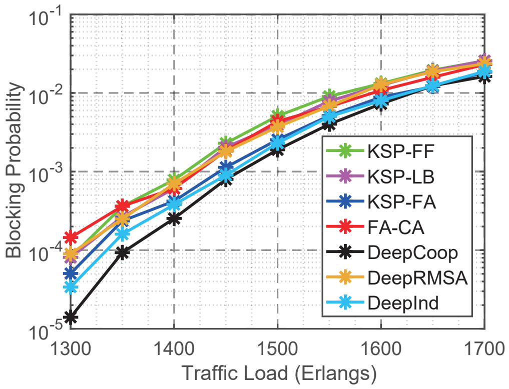


Fig. 8. Results on blocking probability (9-domain SD-EON).

almost unchanged throughout the process, which is expected, while those from the DRL-based approaches decrease with the number of served requests. It can be seen that the blocking probability from DeepCoop converges much faster than that from DeepRMSA, and is the lowest in Fig. 7 after DeepCoop being trained with 2×10^4 requests. On the other hand, DeepRMSA can only slightly outperform KSP-LB after being trained with more than 10^5 requests. This is because we design the actions of the DRL agents in DeepCoop as RSA heuristics, which contributes to a much smaller and more relevant action space than that of DeepRMSA. The blocking probability from DeepCoop converges to $\sim1.85 \times 10^{-3}$ after 6×10^4 requests, and compared with the heuristics (*i.e.*, KSP-FF, KSP-LB, KSP-FA and FA-CA), it achieves 53.8%, 60%, 40%, and 49.8% reduction on blocking probability, respectively.

We notice that the blocking probability of DeepInd converges faster than DeepCoop. This is because the DRL agents of DeepInd is trained independently. Here, in Fig. 7, we can still treat DeepCoop as in its offline training phase before the training processes converge. We also repeat the simulations with various traffic loads, and verify that the results follow the same trend as that in Fig. 7. This confirms that the agents in DeepCoop select RSA heuristics adaptively to serve requests in a dynamic environment, and can achieve better blocking performance than any of the heuristics in its action space.

C. *Performance on Dynamic Multi-Domain Provisioning*

We then provision dynamic lightpath requests in the multi-domain SD-EON with the algorithms. Fig. 8 shows the results on blocking probability, which indicates that DeepCoop provides the lowest blocking probability at all the traffic loads. Specifically, compared with the best deterministic heuristic in Fig. 8 (KSP-FA), DeepCoop reduces the blocking probability 72.25% at most (at $1,300$ Erlangs) and by 23.55% on average. DeepInd performs worse than DeepCoop in Fig. 8, which verifies the benefits of making the DRL agents cooperate with each other. Note that, DeepCoop only makes the DRL agents share limited intra-domain information, and thus it balances the tradeoff between the performance of lightpath provisioning and the autonomy and privacy of domains well.

To further analyze how each agent of DeepCoop selects RSA heuristics adaptively, we record the distributions of selected heuristics in *Domains* 4 and 5 at different loads, and plot them in Figs. 9 and 10, respectively. We can see that the agent in *Domain* 4 prefers KSP-LB and KSP-FF, while the one in *Domain* 5 selects KSP-FA or FA-CA at the highest probability. This is because the topology of *Domain* 4 is relatively simple such that spectrum fragmentation might not be an issue in the dynamic provisioning in it, *i.e.*, the border nodes of *Domain* 4 are usually inter-connected with only one intra-domain link. On the other hand, the topology of *Domain* 5 is much more complex, and thus fragmentation-aware provisioning schemes are more beneficial.

Meanwhile, we notice that even for the same domain, the most selected heuristic can be different at different traffic loads. For instance, when the traffic load increases from $1,350$ Erlangs to $1,550$ Erlangs, the agent in *Domain* 4 changes its most-preferred heuristic from KSP-LB to KSP-FF. This is

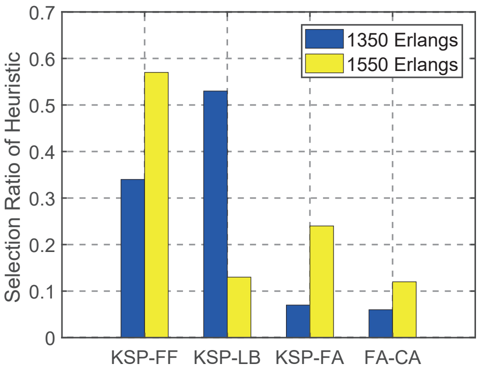


Fig. 9. Distribution of selected heuristics in *Domain* 4.

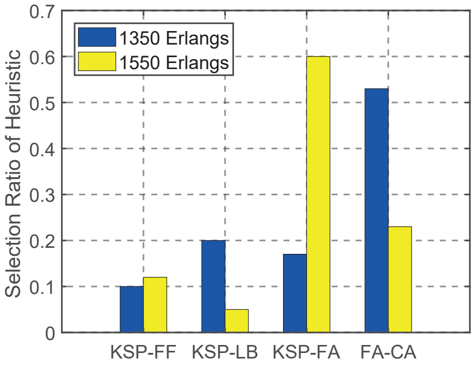


Fig. 10. Distribution of selected heuristics in *Domain* 5.

because even though KSP-LB can balance the spectrum usages in a domain, it provisions lightpaths with a longer average path length than KSP-FF. The longer average path length leads to more spectrum usage per lightpath, which might lead to higher blocking probability when the network is more congested, especially due to the fact that *Domain* 4 sits in the middle of the multi-domain SD-EON and thus has a higher probability of being used as an intermediate domain. To this end, we can see that for inter-domain service provisioning, the best heuristic to use in each domain depends on a few factors (*e.g.*, the topology, traffic load, and position of the domain), and thus the mechanism to choose the right heuristic is rather complicated, especially when the network state can be time-varying. This makes it infeasible to select the heuristics with a deterministic algorithm. However, DeepCoop can leverage DRL to tackle the complicated problem, and it always uses the suitable RSA heuristic to minimize the blocking probability.

D. Evaluations on Universality and Scalability

In order to show the universality of the design of DeepCoop, we change the topology of the multi-domain SD-EON to that in Fig. 11 and redo the simulations. This time, the topology only contains three domains, but each domain is generally larger than those in the 9-domain topology in Fig. 5. Except for the necessary minor changes to adapt to the new topology, we do not change anything in DeepCoop and apply it directly to the inter-domain provisioning in the 3-domain SD-EON. Fig. 12 shows the evolution of the blocking probability in the training when the number of served requests increases (*i.e.*, the traffic load is fixed at $1,300$ Erlangs). It can be seen that DeepCoop providers lower blocking probabilities than all the heuristics after being trained with 2×10^4 requests, and its training converges after 6×10^4 requests. Specifically, if we compare the results in Figs. 7 and 12, we can see that even though the multi-domain SD-EONs use completely different topologies, the training of the DeepCoop in them performs similarly. This confirms the universality and scalability of the distributed online training implemented in DeepCoop. Fig. 13 plots the results on blocking probability in the 3-domain SD-EON, which follow the similar trend as that in Fig. 8.

TABLE III
AVERAGE RUNNING TIME PER INTER-DOMAIN LIGHTPATH PROVISIONING (MILLISECONDS)

Algorithm	KSP-FF	KSP-LB	KSP-FA	FA-CA
3-domain	2.72	3.51	4.20	3.60
9-domain	6.45	7.23	7.68	7.85
Algorithm	DeepCoop	DeepRMSA	DeepInd	-
3-domain	8.88	4.29	7.32	-
9-domain	10.43	8.27	9.65	-

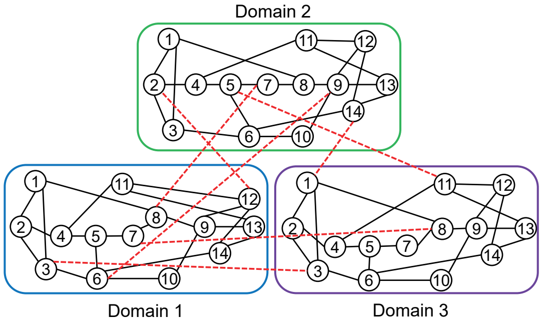


Fig. 11. Topology of 3-domain SD-EON used in simulations.

Table III lists the average running time per inter-domain lightpath provisioning of the algorithms. Here, for all the DRL models (*i.e.*, DeepCoop, DeepRMSA and DeepInd), the running time is only for their online operation/training, because the offline training should be finished before they can be put into operation (*i.e.*, the running time of the offline training does not affect the time-efficiency of the online operation/training). The results in Table III suggest that, similar to the deterministic heuristics, DeepCoop only uses a few milliseconds to serve each inter-domain lightpath request. This confirms that it is suitable for dynamic provisioning. Meanwhile, we notice that due to the complexity of the analysis in each DRL agent, DeepCoop generally takes more time than the deterministic heuristics to provision each inter-domain lightpath. On the other hand, when the number of SD-EON domains increases from 3 to 9, the running time of the deterministic heuristics generally doubles, while that of DeepCoop only increases slightly. This verifies the scalability of DeepCoop.

VI. CONCLUSION

In this paper, we designed and optimized DeepCoop, which is an inter-domain service framework that utilizes multiple

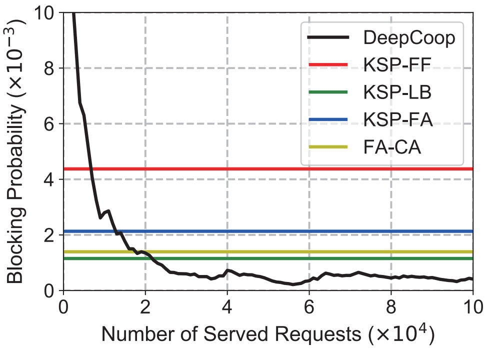


Fig. 12. Evolving of blocking probability in 3-domain SD-EON (traffic load fixed at $1,300$ Erlangs).

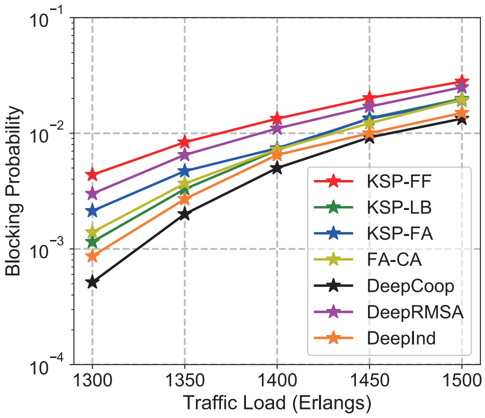


Fig. 13. Results on blocking probability (3-domain SD-EON).

cooperative DRL agents to achieve scalable network automation in a multi-domain SD-EON. Specifically, DeepCoop employs a DRL agent in each domain to optimize intra-domain service provisioning, while a domain-level PCE is introduced to calculate the sequence of the domains to go through for each lightpath request. By sharing a restricted amount of information among each other, the DRL agents can make their decisions distributedly. To ensure scalability and universality, we designed the action space of each DRL agent based on well-known RSA heuristics, and architected the agents based on the SAC scenario. With extensive simulations, we demonstrated that DeepCoop can analyze the network environment in a multi-domain SD-EON to always select the best RSA heuristic to minimize the blocking probability, it outperforms the existing algorithms on inter-domain provisioning in various simulation scenarios, and the distributed training implemented in it ensures its universality and scalability (*i.e.*, its training and operation do not depend on the topology of the SD-EON).

Acknowledgments

This work was supported in part by the NSFC project 61871357, Zhejiang Lab Research Fund 2019LE0AB01, SPR Program of CAS (XDC02070300), and Fundamental Funds for Central Universities (WK3500000006).

References

[1] P. Lu *et al.*, "Highly efficient data migration and backup for Big Data applications in elastic optical inter-data-center networks," *IEEE Netw.*, vol. 29, pp. 36–42, Sept./Oct. 2015.

[2] H. Lu, M. Zhang, Y. Gui, and J. Liu, "QoE-driven multi-user video transmission over SM-NOMA integrated systems," *IEEE J. Sel. Areas Commun.*, vol. 37, pp. 2102–2116, Sept. 2019.

[3] L. Gong, H. Jiang, Y. Wang, and Z. Zhu, "Novel location-constrained virtual network embedding (LC-VNE) algorithms towards integrated node and link mapping," *IEEE/ACM Trans. Netw.*, vol. 24, pp. 3648–3661, Dec. 2016.

[4] H. Wu and H. Lu, "Delay and power tradeoff with consideration of caching capabilities in dense wireless networks," *IEEE Trans. Wireless Commun.*, vol. 18, pp. 5011–5025, Oct. 2019.

[5] J. Liu *et al.*, "On dynamic service function chain deployment and readjustment," *IEEE Trans. Netw. Serv. Manag.*, vol. 14, pp. 543–553, Sept. 2017.

[6] Y. Gui, H. Lu, F. Wu, and C. Chen, "Robust video broadcast for users with heterogeneous resolution in mobile networks," *IEEE Trans. Mobile Comput.*, *in Press*, 2020.

[7] Z. Zhu, W. Lu, L. Zhang, and N. Ansari, "Dynamic service provisioning in elastic optical networks with hybrid single-/multi-path routing," *J. Lightw. Technol.*, vol. 31, pp. 15–22, Jan. 2013.

[8] L. Gong and Z. Zhu, "Virtual optical network embedding (VONE) over elastic optical networks," *J. Lightw. Technol.*, vol. 32, pp. 450–460, Feb. 2014.

[9] M. Zeng, W. Fang, and Z. Zhu, "Orchestrating tree-type VNF forwarding graphs in inter-DC elastic optical networks," *J. Lightw. Technol.*, vol. 34, pp. 3330–3341, Jul. 2016.

[10] N. McKeown *et al.*, "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, pp. 69–74, Mar. 2008.

[11] S. Li *et al.*, "Protocol oblivious forwarding (POF): Software-defined networking with enhanced programmability," *IEEE Netw.*, vol. 31, pp. 12–20, Mar./Apr. 2017.

[12] R. Casellas *et al.*, "Control and management of flexi-grid optical networks with an integrated stateful path computation element and OpenFlow controller," *J. Opt. Commun. Netw.*, vol. 5, pp. A57–A65, Oct. 2013.

[13] C. Chen *et al.*, "Demonstrations of efficient online spectrum defragmentation in software-defined elastic optical networks," *J. Lightw. Technol.*, vol. 32, pp. 4701–4711, Dec. 2014.

[14] X. Chen *et al.*, "Flexible availability-aware differentiated protection in software-defined elastic optical networks," *J. Lightw. Technol.*, vol. 33, pp. 3872–3882, Sept. 2015.

[15] R. Munoz *et al.*, "Transport network orchestration for end-to-end multilayer provisioning across heterogeneous SDN/OpenFlow and GMPLS/PCE control domains," *J. Lightw. Technol.*, vol. 33, pp. 1540–1548, Apr. 2015.

[16] S. Das, G. Parulkar, and N. McKeown, "Why OpenFlow/SDN can succeed where GMPLS failed," in *Proc. of ECOC 2012*, pp. 1–3, Sept. 2012.

[17] Z. Zhu *et al.*, "OpenFlow-assisted online defragmentation in single-/multi-domain software-defined elastic optical networks," *J. Opt. Commun. Netw.*, vol. 7, pp. A7–A15, Jan. 2015.

[18] L. Sun, X. Chen, and Z. Zhu, "Multi-broker based service provisioning in multi-domain SD-EONs: Why and how should the brokers cooperate with each other?" *J. Lightw. Technol.*, vol. 35, pp. 3722–3733, Sept. 2017.

[19] R. Proietti *et al.*, "Experimental demonstration of machine-learning-aided QoT estimation in multi-domain elastic optical networks with alien wavelengths," *J. Opt. Commun. Netw.*, vol. 11, pp. A1–A10, Jan. 2019.

[20] K. Christodoulopoulos, I. Tomkos, and E. Varvarigos, "Elastic bandwidth allocation in flexible OFDM-based optical networks," *J. Lightw. Technol.*, vol. 29, no. 9, pp. 1354–1366, May 2011.

[21] B. Chatterjee, N. Sarma, and E. Oki, "Routing and spectrum allocation in elastic optical networks: A tutorial," *IEEE Commun. Surveys Tuts.*, vol. 17, pp. 1776–1800, Third Quarter 2015.

[22] A. Castro *et al.*, "Brokered orchestration for end-to-end service provisioning across heterogeneous multi-operator (multi-AS) optical networks," *J. Lightw. Technol.*, vol. 34, pp. 5391–5400, Dec. 2016.

[23] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[24] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 2018. [Online]. Available: http://incompleteideas.net/book/first/the-book.html

[25] H. Fang *et al.*, "Predictive analytics based knowledge-defined orchestration in a hybrid optical/electrical datacenter network testbed," *J. Lightw. Technol.*, vol. 37, pp. 4921–4934, Oct. 2019.

[26] W. Lu *et al.*, "AI-assisted knowledge-defined network orchestration for energy-efficient data center networks," *IEEE Commun. Mag.*, vol. 58, pp. 86–92, Jan. 2020.

[27] B. Li, W. Lu, and Z. Zhu, "Deep-NFVOrch: Leveraging deep reinforcement learning to achieve adaptive vNF service chaining in EON-DCIs," *J. Opt. Commun. Netw.*, vol. 12, pp. A18–A27, Jan. 2020.

[28] X. Chen *et al.*, "DeepRMSA: A deep reinforcement learning framework for routing, modulation and spectrum assignment in elastic optical networks," *J. Lightw. Technol.*, vol. 37, pp. 4155–4163, Aug. 2019.

[29] X. Chen *et al.*, "Multi-agent deep reinforcement learning in cognitive inter-domain networking with multi-broker orchestration," in *Proc. of OFC 2019*, pp. 1–3, Mar. 2019.

[30] B. Li and Z. Zhu, "DeepCoop: Leveraging cooperative DRL agents to achieve scalable network automation for multi-domain SD-EONs," in *Proc. of OFC 2020*, pp. 1–3, Mar. 2020.

[31] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft Actor-Critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," *arXiv preprint arXiv:1801.01290*, Aug. 2018. [Online]. Available: http://arxiv.org/abs/1801.01290.

[32] L. Gong, X. Zhou, W. Lu, and Z. Zhu, "A two-population based evolutionary approach for optimizing routing, modulation and spectrum assignments (RMSA) in O-OFDM networks," *IEEE Commun. Lett.*, vol. 16, pp. 1520–1523, Sept. 2012.

[33] Y. Yin *et al.*, "Spectral and spatial 2D fragmentation-aware routing and spectrum assignment algorithms in elastic optical networks," *IEEE J. Opt. Commun. Netw.*, vol. 5, pp. A100–A106, Oct. 2013.

[34] H. Wu, F. Zhou, Z. Zhu, and Y. Chen, "On the distance spectrum assignment in elastic optical networks," *IEEE/ACM Trans. Netw.*, vol. 25, pp. 2391–2404, Aug. 2017.

[35] L. Gong *et al.*, "Efficient resource allocation for all-optical multicasting over spectrum-sliced elastic optical networks," *J. Opt. Commun. Netw.*, vol. 5, pp. 836–847, Aug. 2013.

[36] L. Yang *et al.*, "Leveraging light-forest with rateless network coding to design efficient all-optical multicast schemes for elastic optical networks," *J. Lightw. Technol.*, vol. 33, pp. 3945–3955, Sept. 2015.

[37] Z. Zhu *et al.*, "Impairment- and splitting-aware cloud-ready multicast provisioning in elastic optical networks," *IEEE/ACM Trans. Netw.*, vol. 25, pp. 1220–1234, Apr. 2017.

[38] L. Zhang and Z. Zhu, "Spectrum-efficient anycast in elastic optical inter-datacenter networks," *Opt. Switch. Netw.*, vol. 14, pp. 250–259, Aug. 2014.

[39] W. Shi, Z. Zhu, M. Zhang, and N. Ansari, "On the effect of bandwidth fragmentation on blocking probability in elastic optical networks," *IEEE Trans. Commun.*, vol. 61, pp. 2970–2978, Jul. 2013.

[40] W. Lu *et al.*, "Implementation and demonstration of revenue-driven provisioning for advance reservation requests in OpenFlow-controlled SD-EONs," *IEEE Commun. Lett.*, vol. 18, pp. 1727–1730, Oct. 2014.

[41] S. Li, W. Lu, X. Liu, and Z. Zhu, "Fragmentation-aware service provisioning for advance reservation multicast in SD-EONs," *Opt. Express*, vol. 23, pp. 25 804–25 813, Oct. 2015.

[42] B. Zhao, X. Chen, J. Zhu, and Z. Zhu, "Survivable control plane establishment with live control service backup and migration in SD-EONs," *J. Opt. Commun. Netw.*, vol. 8, pp. 371–381, Jun. 2016.

[43] M. Fabrega *et al.*, "Demonstration of adaptive SDN orchestration: A real-time congestion-aware services provisioning over OFDM-based 400G OPS and flexi-WDM OCS," *J. Lightw. Technol.*, vol. 35, pp. 506–512, Jan. 2017.

[44] Z. Zhu *et al.*, "Demonstration of cooperative resource allocation in an OpenFlow-controlled multidomain and multinational SD-EON testbed," *J. Lightw. Technol.*, vol. 33, pp. 1508–1514, Apr. 2015.

[45] D. Marconett, L. Liu, and B. Yoo, "Optical FlowBroker: load-balancing in software-defined multi-domain optical networks," in *Proc. of OFC 2014*, pp. 1–3, Mar. 2014.

[46] D. Marconett and B. Yoo, "FlowBroker: Market-driven multi-domain SDN with heterogeneous brokers," in *Proc. of OFC 2015*, pp. 1–3, Mar. 2015.

[47] X. Chen *et al.*, "Incentive-driven bidding strategy for brokers to compete for service provisioning tasks in multi-domain SD-EONs," *J. Lightw. Technol.*, vol. 34, pp. 3867–3876, Aug. 2016.

[48] P. Vamvakas, E. Tsiropoulou, and S. Papavassiliou, "Dynamic spectrum management in 5G wireless networks: A real-life modeling approach," in *Proc. of INFOCOM 2019*, pp. 1–9, Apr. 2019.

[49] Q. Yao, H. Yang, A. Yu, and J. Zhang, "Transductive transfer learning-based spectrum optimization for resource reservation in seven-core elastic optical networks," *J. Lightw. Technol.*, vol. 37, pp. 4164–4172, Aug. 2019.

[50] J. Yu *et al.*, "A deep learning based RSA strategy for elastic optical networks," in *Proc. of ICOCN 2019*, pp. 1–3, Aug. 2019.

[51] G. Liu *et al.*, "Hierarchical learning for cognitive end-to-end service provisioning in multi-domain autonomous optical networks," *J. Lightw. Technol.*, vol. 37, pp. 218–225, Jan. 2019.

[52] F. Paolucci *et al.*, "A survey on the path computation element (PCE) architecture," *IEEE Commun. Surveys Tut.*, vol. 15, pp. 1819–1841, Fourth Quarter 2013.

[53] R. Lowe *et al.*, "Multi-agent actor-critic for mixed cooperative-competitive environments," *arXiv:1706.02275*, Jun. 2017. [Online]. Available: http://arxiv.org/pdf/1706.02275v4.

[54] Y. Yang *et al.*, "Mean field multi-agent reinforcement learning," *arXiv:1802.05438*, Jul. 2018. [Online]. Available: http://arxiv.org/abs/1802.05438.

[55] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," *arXiv:1802.09477*, Oct. 2018. [Online]. Available: http://arxiv.org/abs/1802.09477.

[56] H. Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," *arXiv:1509.06461*, Dec. 2015. [Online]. Available: http://arxiv.org/abs/1509.06461.

[57] D. Kingma and J. Ba, "Adam: A method for stochastic optimization (v9)," *arXiv:1412.6980*, 2017. [Online]. Available: https://arxiv.org/abs/1412.6980.

[58] J. Zhao, S. Subramaniam, and M. Brandt-Pearce, "Intradomain and interdomain QoT-aware RWA for translucent optical networks," *J. Opt. Commun. Netw.*, vol. 6, pp. 536–548, Jun. 2014.