Journal Paper

Adaptive Network Function Chain Orchestration Strategy For 6G Inclusive Intelligent Services

Yi Wen

Sai Zou

Yanglong Sun

Wei Ni

Kai Li*

Hongfeng Gao

*CISTER Research Centre

CISTER-TR-250101

2025

Adaptive Network Function Chain Orchestration Strategy For 6G Inclusive Intelligent Services

Yi Wen, Sai Zou, Yanglong Sun, Wei Ni, Kai Li*, Hongfeng Gao

*CISTER Research Centre

Polytechnic Institute of Porto (ISEP P.Porto)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8321159

E-mail: wenyiiii@foxmail.com, dr-zousai@foxmail.com, ylsun4work@163.com, Wei.Ni@data61.csiro.au, kai@isep.ipp.pt, hfgao@gzu.edu.cn

https://www.cister-labs.pt

Abstract

In the upcoming 6G era, inclusive intelligent services (IIS) that rely on integrated communications and AI arithmetic will become the norm. These services require efficient distributed intelligent learning or reasoning. However, with the proliferation of complex applications, providing differentiated and customized services through effective Network Function Chain (NFC) orchestration has become a significant challenge. In this paper, an adaptive NFC orchestration strategy (ANFCOS) for 6G IIS is proposed to minimize costs. In particular, a tailored NFC orchestration is optimized, catering to diverse quality of service requirements. ANFCOS models this process as a Markov decision process and multi-actor-attention-critic (MAAC) reinforcement learning method for solving. The strategy enables the deployment and operational cost optimization of multiple parallel NFCs, relying on a dynamic attention-shifting mechanism. Additionally, We introduced a reinforcement learning framework based on maximum entropy to enhance the MAAC for solving and demonstrate its convergence. Simulation results show that ANFCOS significantly improves the network performance of 6G intelligent services. Compared with MADDPG, DDPG and MCTS, the cost of orchestration was reduced by 8.2%, 22.1% and 28.6%; the total delay of orchestration was reduced by 7.7%, 12.7% and 10.8%.

Adaptive Network Function Chain Orchestration Strategy for 6G Inclusive Intelligent Services

Yi Wen, Sai Zou,* *Member, IEEE*, Yanglong Sun,* *Member, IEEE*, Wei Ni, *Fellow, IEEE*, Kai Li, *Senior Member, IEEE*, and Hongfeng Gao

***Abstract*—In the upcoming 6G era, inclusive intelligent services(IISs) that rely on integrated communications and AI arithmetic will become the norm. These services require efficient distributed intelligent learning or reasoning. However, with the proliferation of complex applications, providing differentiated and customized services through effective network function chain (NFC) orchestration has become a significant challenge. In this paper, an adaptive NFC orchestration strategy (ANFCOS) for 6G IIS is proposed to minimize costs. In particular, a tailored NFC orchestration is optimized, catering to diverse quality of service requirements. ANFCOS models this process as a Markov decision process and multi-actor-attention-critic (MAAC) reinforcement learning method for solving. The strategy enables the deployment and operational cost optimization of multiple parallel NFCs, relying on a dynamic attention-shifting mechanism. Additionally, We introduced a reinforcement learning framework based on maximum entropy to enhance the MAAC for solving and demonstrate its convergence. Simulation results show that ANFCOS significantly improves the network performance of 6G intelligent services. Compared with MADDPG, DDPG and MCTS, the cost of orchestration was reduced by 8.2%, 22.1% and 28.6%; the total delay of orchestration was reduced by 7.7%, 12.7% and 10.8%**

***Index Terms*—Inclusive intelligent services, Network function chain, Adaptive orchestration, Dynamic attention mechanism, Multi-Agent reinforcement learning.**

I. INTRODUCTION

Future 6G mobile systems are anticipated to be a transformative evolution, focusing on extending intelligent services to every societal corner, i.e., delivering inclusive intelligent services (IISs) [1]. An IIS originates from an array of intelligent terminals and their collaborative learning. It is characterized by diverse service functions, deep device interconnectivity that leads to real-time dynamic service operations, extensive data processing demands and stringent performance standards for AI tasks. In this context, an adaptive network function chain (NFC) orchestration strategy plays a critical role [2]. NFC sequentially links various network functions in a specific order to form a service chain. An orchestration strategy dynamically allocates network resources in response to real-time network changes, optimizing resource management and adapting to varied service demands to meet rigorous performance metrics. [3]. In the context of 6G IISs, this strategy enhances network adaptability for extensive data processing demands, and guarantees the quality-of-service (QoS) of these services. Combining adaptive NFC orchestration with edge computing improves data handling and transmission [4].

Background: With the widespread application of 6G IISs, the network will connect billions of terminals, leading to an explosive growth in network equipment and services demands [1]. Quickly sensing network changes and responding to user needs from different terminals is essential but challenging [5]. At the same time, guaranteeing QoS and reducing orchestration costs through efficient resource management complicates this challenge. However, it is a vital objective for cloud service providers (CSPs). Therefore, orchestrating complex NFCs to cater to the needs of IISs continues to pose a substantial and intriguing challenge [6], [7].

Challenge: The integration of network function virtualization (NFV), software-defined networking (SDN), and edge cloud technology allows traditional network functions to be deployed as software on servers [8]. NFC orchestration strategy uses SDN to link virtual network functions (VNFs) in sequence to meet service needs, enabling flexible resource management and monitoring [9]. VNF-sharing technology further enhances resource utilization and reduces costs by allowing shared instances [10]. However, centralized VNF deployment lowers costs but risks resource strain and degraded QoS, while distributed deployment balances load at the cost of higher deployment expenses [11], [12].

In 6G networks, collaboration across different network function layers and resource domains is crucial, with VNFs often having dependency or mutual-exclusion relationships, such as protocol interdependencies (e.g., TCP/IP, HTTP and FTP) and conflicting functions like firewalls and remote login [13]. Adaptive NFC orchestration strategies face the following

This work is supported by the National Natural Science Foundation of China (62361011, 62462012, U24A20246), the Natural Science Foundation of Guizhou (ZK[2022]-162), the Social Science Foundation of China (21VSZ126), the Natural Science Foundation of Guizhou University (X2021167), and the Guizhou Cloud Network Collaborative Innovation Center, and the Guizhou Cloud Network Collaborative Deterministic Transmission Engineering center ([2023]032), and the Guizhou Engineering Research Center for smart services (2203-520102-04-04-298868), and the CISTER Research Unit (UIDP/UIDB/04234/2020) and project ADANET (PTDC/EEICOM/3362/2021), financed by National Funds through FCT/MCTES (Portuguese Foundation for Science and Technology).

Part of this paper was presented at the ACM MobiCom '24.

Yi Wen and Sai Zou and Hongfeng Gao are with the College of Big Data and Information Engineering, Guizhou University (e-mail: wenyiiii@foxmail.com; dr-zousai@foxmail.com).

Yanglong Sun is with Navigation College, Jimei University (e-mail: ylsun4work@163.com).

Wei Ni is with the Data61 Business Unit, CSIRO (e-mail: Wei.Ni@data61.csiro.au).

Kai Li is with the Division of Electrical Engineering, Department of Engineering, University of Cambridge, CB3 0FA Cambridge, U.K., and also with the Real-Time and Embedded Computing Systems Research Centre (CISTER), Porto 4249-015, Portugal (e-mail: kaili@ieee.org).

Sai Zou and Yanglong Sun are co-corresponding authors.

challenges:

1) Resource management: Resource management remains a crucial challenge in the orchestration of NFC, not only allocating resources efficiently but also addressing the delay constraints of requests and the resource limitations of devices and links [14]. How to efficiently reuse VNF instances and fully leverage the idle resources of devices in such extensive networks poses significant challenges.
2) Cost optimization: Cost reduction serves as a primary incentive for CSPs to provide IISs. With VNF sharing technology, some existing methods favor centralized deployment to minimize costs but risk congestion [15], while distributing VNFs eases device pressure but can cause redundancy [16]. How to balance cost and load remains a key challenge.
3) Cross-domain and cross-layer collaboration: Different network functional layers and resource domains introduce dependencies and conflicts among VNFs within an NFC [17]. How to describe the implicit relations between NFCs is still a challenge to complete 6G IISs in support of cross-domain and cross-layer collaboration.

Motivation: Recently, deep reinforcement learning (DRL) has demonstrated remarkable potential in addressing NFC orchestration problems [18]. A notable branch, multi-agent DRL (MADRL), excels at facilitating agent collaboration to achieve shared objectives, making it particularly effective for tackling complex VNF layout challenges [19]. In addition, there have been studies integrating attention mechanisms in DRL for solving the NFC orchestration problem, in terms of decision accuracy and speed and showing significant improvements [20]. Inspired by these advancements, we explore the application of MADRL combined with attention mechanisms to enhance the accuracy and efficiency of NFC orchestration in complex environments.

Contributions: This paper introduces an innovative adaptive NFC orchestration strategy (ANFCOS) for 6G IIS, designed to reduce operational costs and minimize end-to-end latency in service requests. Utilizing a multi-actor-attention-critic (MAAC) [21] framework within an edge cloud context, it employs an attention mechanism to focus on key information, revealing hidden relations between NFCs across different network function layers or different network resource domains, thus improving computational efficiency. ANFCOS models the orchestration of NFCs as a Markov decision process (MDP), thereby facilitating improved decision-making accuracy and collaborative efficiency. The key contributions of this paper are:

1) We study the problem of adaptive NFC dynamic orchestration of IIS across different network function layers and resource domains in an edge cloud environment and formulate it as an MDP problem.
2) We propose an adaptive MAAC-based NFC dynamic orchestration strategy (ANFCOS) to guarantee low cost and QoS. ANFCOS integrates an attention mechanism to capture the association between NFCs, thereby improving its the accuracy and efficiency.
3) We propose an action strategy optimization algorithm. It optimizes the action space of the algorithm and reduces the computational complexity of MAAC, making ANFCOS more adaptable to the NFC orchestration problem in large-scale networks.
4) We integrate the soft actor-critic (SAC) algorithm into the MAAC framework to enhance the search capability, and provide it with a detailed mathematical analysis to demonstrate the reliable convergence of the ANFCOS method.

We conduct extensive simulations and compare them with the state-of-the-art approaches in the literature to demonstrate the excellent performance of our proposed ANFCOS. These simulations confirm that ANFCOS significantly exceeds these benchmarks, compared with MADDPG, DDPG and MCTS, ANFCOS reduces costs by 8.2%, 22.1% and 28.6%, and orchestrates end-to-end delays by 7.7%, 12.7% and 10.8% respectively.

The remainder of this paper is organized as follows. Section II summarizes related work. Section III introduces system model, including scenario description and formulation of the NFC orchestration problem. Section IV introduces ANFCOS in detail. Section V presents our simulation results compared with other methods. Section VI gives the conclusion.

II. RELATED WORK

In this section, we briefly review representative NFC orchestration methods, attention mechanisms, and DRL. We also explain the correlation between the above relevant methods and this paper.

A. NFC Orchestration Strategies

The study of NFC orchestration issues has become a hot research topic [22]. However, these studies have paid little attention to the impact of VNF sharing mechanisms on different policies and the interoperability of different VNF types in cross-layer, cross-domain deployments.

Xu et al. [23] proposed an adaptive NFC deployment mechanism for the Internet of Things, leveraging deep Q-networks (DQN) and tidal virtual machine (VM) control for scalable VM management to optimize resource utilization while meeting service requirements. Cao et al. [24] introduced an embedding and scheduling method for SRs to meet maximum delay constraints by perceiving physical device resource capabilities. Yue et al. [15] developed a heuristic throughput optimization and latency assurance (TO-DG) scheme, incorporating affinity-based service function chain mapping and VNF tuning algorithms. Bari et al. [25] analyzed VNF instance optimization and placement to enhance operational efficiency and cost-effectiveness while maintaining compliance with service level agreements (SLAs) and device-specific preferences. Xu et al. [26] explored NFC orchestration in cloud-edge collaborative environments, proposing a multi-objective NFC deployment model to balance industrial internet of things (IIoT) service quality with resource utilization, addressing diverse service demands and network complexities.

It is noteworthy that the above-mentioned studies, while addressing the problem of VNF deployment in NFC coordination, lack consideration for the possibility that NFC may not be deployed on the shortest path following VNF sharing, and the issue of whether VNFs are centralized or dispersed across the network. Discussions on the heterogeneity among VNFs in cross-layer and cross-domain issues are limited. In contrast, this paper discusses the close connection between devices and VNF deployment from both perspectives.

B. DRL Methods

DRL has proven effective for dynamic combinatorial optimization [27], but it faces challenges in large-scale networks due to the need for frequent interaction, new data collection, and retraining to adapt to diverse service requests, slowing down training and complicating traditional DRL approaches.

Researchers have proposed DRL-based methods for online NFC deployment. Haeri et al. [28] modeled the virtual node embedding problem as an MDP and employed monte carlo tree search (MCTS) to maximize infrastructure provider profits, though its high complexity requires multiple passes per request. However, this algorithm requires multiple passes to find the solution for each request, which requires high complexity. Gu et al. [29] developed a DDPG-based algorithm to maximize network utility by considering edge computing and NFC orchestration costs, but DDPG's computational demands limit its real-time applicability. However, as DDPG is a computationally intensive algorithm, it may not meet the real-time requirements in scenarios where rapid response is needed. MARL, which decomposes large tasks into smaller sub-tasks [30], has become popular for NFC orchestration due to its ability to address NP-hard problems [31]. Zhu et al. [32] introduced a MADDPG algorithm for optimizing NFC deployment.

However, these methods often overlook the influence of surrounding nodes' resources on network state. DRL models can differentiate the importance of neighboring nodes based on their available resources, while attention mechanisms further enhance resource selection and facilitate efficient neighbor interactions [33].

C. Attention Mechanisms

Attention mechanisms have been widely used in machine learning tasks due to their adaptability and effectiveness in modeling dependencies. In recent deep reinforcement learning research, attention mechanisms have been applied to address scalability issues caused by dynamic agent interactions, and facilitate information sharing between agents, thereby enhancing their cooperative capabilities.

Chen et al. [34] used Transformers to preprocess input dimensions in reinforcement learning, addressing the issue of dynamic agent numbers. While effective in adapting to varying input sizes, this approach requires costly pre-training and does not address multi-agent collaboration challenges. Xue et al. [35] applied Transformers in the PPO algorithm toenables the sharing of state information among all the agents via Transformer, but it is limited to cooperative environments where agents share all parameters. Unlike the existing studies, our considered scenario involves agents competing for resources while collaborating on shared goals. Li et al. [36] introduced a collaborative multi-agent deep reinforcement learning (CM-DRL) algorithm, which shows strong efficiency and stability in multi-objective balancing. our considered scenario, cost optimization focuses on resource deployment through NFC, rather than balancing multiple objectives. Therefore, our study concentrates on optimizing the cost of deploying the network function chain and does not address the trade-off between multiple optimization objectives.

Attention mechanisms have also been increasingly used in deep reinforcement learning, but most studies focus on data processing before training or purely cooperative environments. In NFV orchestration, however, NFVs both cooperate and compete for resources. To address this, our method integrates an attention mechanism into the Q-value calculation of the critic network. By evaluating the state and action information of other agents, it improves individual decisions and optimizes the overall objectives under competitive settings.

Building on the insights from previous research, we delve into the NFC orchestration problem within IIS contexts. Our focus lies in optimizing VNF deployment costs in NFC orchestration, taking into account the association between VNFs and devices. We then implement adaptive orchestration strategies using the MAAC approach in MARL. The input space of the MAAC method scales linearly with the number of agents, rather than quadratically, enabling effective handling of a multitude of device requests in inclusive intelligent scenarios. This method incorporates an attention model among NFCs, allowing for the efficient identification and utilization of crucial information pertinent to current NFC deployment decisions, thereby optimizing deployment efficiency.

III. SYSTEM MODEL

In this section, the application scenario is delineated, where the physical architecture of the edge cloud platform with the model for service requests is presented. Subsequently, we proceed to qualify the mapping rules between VNFs and devices under the cross-layer and cross-domain configuration of the cross-network system. Concurrently, the underlying interconnections among VNF configurations are formulated into the attention mechanism of the NFC. Culminating in the development of a mathematical model, this framework addresses the orchestration challenges inherent in NFC. TABLE I lists the notation used in the paper.

A. Scenario Description

This paper investigates a tripartite network architecture depicted in Fig. 1. The apex tier, known as the application layer, facilitates access to 6G IISs and seamlessly transitions these into SRs. This process enables the elucidation of both the requisites for requested resources and the demands for VNFs. The nadir tier, or the edge cloud platform, executes the deployment of VNFs in response to control commands. Furthermore, it orchestrates these VNFs in accordance with the sequence prescribed by the service request (SR), thereby constituting an NFC. Each server within the network is capable of hosting one or multiple distinct VNFs; moreover, a single VNF has the capacity to process multiple requests originating

TABLE I: Notation and Definition

Notation	Definition
G	The network weighted undirected graph
N	The set of device nodes in network graph
L	The set of links in network graph
n	The device node in network graph, $n \in N$
l_{nm}	The link in network graph, $l_{nm} \in L$
C_n^{com},C_n^{mem}	The computation, memory resource capacity on device n
C_{nm}^{bw}	The bandwidth of link l_{nm}
$\mathcal{F}$	The set of SR
f	The index for each SR, $f \in \mathcal{F}$
V_f	The set of VNF in SR f
L_f	The set of virtual link in f
v_f	The v-th VNF in SR f
$l_{u_f v_f}$	The virtual link between v_f and u_f
$R_{v_f}^{\text{com}}$, $R_{v_f}^{\text{mem}}$	The computation and memory resource demand of VNF v_f
$R_{l_{u_f v_f}}^{\text{bw}}$	The bandwidth resource demand of SR f
$\mathcal{P}$	The set of all VNF type
p	The index of type VNF
$v_{f,p}$	The type of v-th VNF in SR f

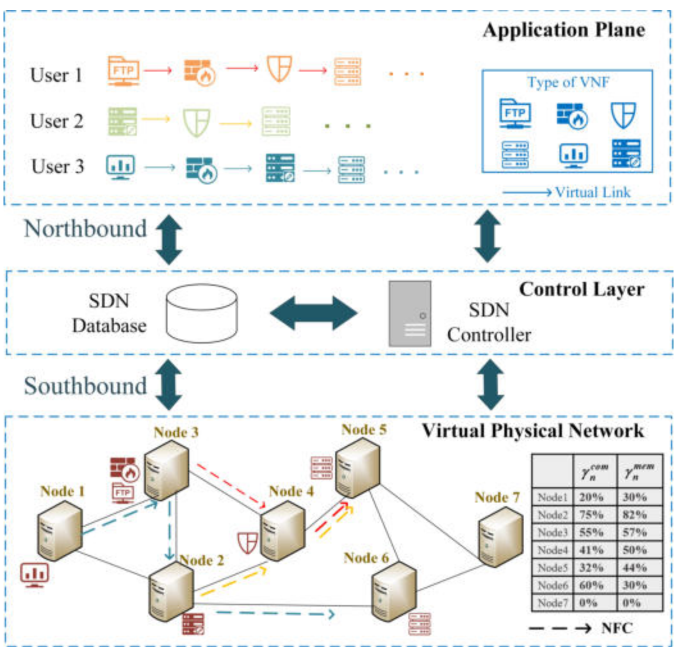


Fig. 1: Three-layer structure of SDN. γ_n^{com} and γ_n^{mem} represent the utilization of the computing and storage resources, respectively. Dotted arrows represent data flow rules between servers.

from VNFs of identical classification. The intermediary stratum serves as the control layer, outfitted with both northbound and southbound interfaces. Through the southbound interface, this layer establishes a connection with the application layer, thereby facilitating the conveyance of service request directives. Concurrently, via the northbound interface, it interfaces with the edge cloud platform. This connection enables the aggregation of essential network state data and the transmission of control commands, which are pivotal for the meticulous orchestration of NFC.

B. Physical Network Model

The network topology is mathematically modeled by $G = (N, L)$, where N delineates the set of physical servers and L represents the ensemble of network links. It is posited that within this network exist physical nodes n and m, such that $n, m \in N$. The direct physical connection between these nodes is symbolized by l_{nm}, with $l_{nm} \in L$, encapsulating the relationship between any two given nodes within the network framework.

Regarding network resources, denote C_n^{com}, C_n^{mem} as the computational capacity and memory capacity of device n respectively, and C_{nm}^{bw} as the bandwidth resources of link l_{nm}. Moreover, let γ_n^{com}, γ_n^{mem}, and γ_{nm}^{bw} to indicate the resource utilization rates of computational capacity, memory capacity and bandwidth capacity, respectively.

C. Service Request Model

Service requests (SRs) are manifested within the network as NFCs, denoted the set of SRs as $\mathcal{F}$, where each SR is indexed by $f \in \mathcal{F} = \{1, 2, \ldots, F\}$. Each request f has VNFs set V_f and virtual link set L_f that need to be deployed. It is posited that within this SR exist VNFs v_f, u_f, such that $v_f, u_f \in V_f$. The virtual connection between these VNFs is symbolized by $l_{u_f v_f}$, with $l_{u_f v_f} \in L_f$.

Regarding resources requirements of SR f, denote $R_{v_f}^{\text{com}}$, $R_{v_f}^{\text{mem}}$ as the computational and memory resource requirements of the VNF v_f, respectively. Additionally, $R_{u_f v_f}^{\text{bw}}$ signifies the bandwidth requirement for the virtual link $l_{u_f v_f}$. Each request is associated with a maximum tolerable delay T_f. Furthermore, let the total VNF types set be $\mathcal{P}$, and each type of VNF is indexed by $p \in \mathcal{P} = \{1, 2, \ldots, P\}$, $v_{f,p}$ indicates that the VNF type of v_f is p.

D. Correspondence Rule Model between VNFs and Devices

To manage VNFs with dependencies and exclusions interact across different network function layers and resource domains, we've designed a rule for VNF deployment. In instances where VNFs of types p and q, with $p, q \in \mathcal{P}$, exhibit mutual exclusion and thus cannot coexist on the same device, a binary variable $z_{p,q}$ is utilized to delineate this constraint. Where, $z_{p,q} = 0$ means types p and q VNFs can share a device, while $z_{p,q} = 1$ indicates they cannot. $z_{p,q}$ helps us determine if a device n can host a type p VNF, using another variable, $\delta_{n,p}$.

$$\delta_{n,p} = \begin{cases} 1 & \text{if } \forall q \in \mathcal{P}, q \neq p : z_{p,q} \cdot y_{n,q} = 0 \\ 0 & \text{otherwise} \end{cases}. \tag{1}$$

where $y_{n,q}$ is a binary variable, $y_{n,q} = 1$ if the device n has deployed q-type VNF, otherwise, $y_{n,q} = 0$. Consequently, when $\delta_{n,p} = 1$ implies that the device n can deploy VNFs of type p without any conflict. Otherwise $\delta_{n,p} = 0$, it indicates that deploying VNFs of type p on device n is not feasible, as it would result in a conflict with other VNFs already deployed within the network.

E. Attention Model between NFCs

Due to the cross-domain and cross-layer collaboration, conflicts arise when deploying different types of VNFs on devices. By establishing an attention mechanism between NFCs, we can enable the network to adaptively identify those VNFs that have the most significant impact on current network performance, thereby resolving deployment conflicts more efficiently and optimizing the collaboration between NFCs. The attention calculation formula between f and f' is denoted as $Att_{f,f'}$:

$$Att_{f,f'} = \mathrm{softmax}\left(\frac{W_q e_f e_{f'}^T W_K^T}{\sqrt{d_K}}\right) W_V e_{f'}, \tag{2}$$

where e_f and $e_{f'}$ embody the joint vector representations that articulate the VNF deployment Strategy and the respective network states about NFC f and f'. These concepts are elaborated upon in IV-C. projection W linearly transforms the features of the input sequence, $W_q \in \mathbb{R}^{|e_f| \times d_K}$, $W_K \in \mathbb{R}^{|e_{f'}| \times d_K}$, $W_V \in \mathbb{R}^{|e_{f'}| \times d_K}$, W_q transforms e_f into "query", W_K transforms $e_{f'}$ into "key", and W_V transforms $e_{f'}$ into "value". For any two NFCs, this attention weight determines the extent of influence or relevance they have on each other.

F. Mathematical Model for NFC Orchestration

The goal of the ANFCOS method is to minimize the operational costs associated with the joint optimization of VNF deployment and NFC embedding, while also reducing the end-to-end latency post-NFC embedding. We define the deployment cost M and resource cost Φ separately. The deployment cost M varies with the type of VNF, denoted V_p^{cost} as the deployment cost of p-type VNF. The formula for calculating M is given by

$$M = \sum_{n \in N} V_p^{\mathrm{cost}} \cdot \delta_{n,p} \cdot y_{n,p}. \tag{3}$$

Resource costs arise from using resources in both nodes and links. Nodes balance network loads by reasonable distribution resources to prevent bottlenecks. Simply put, using more resources leads to higher costs. When resource utilization is low, it's more cost-effective to reuse VNF instances. Therefore, when resource utilization is high the costs increase rapidly, deploying new VNF instances becomes more economical than reusing them. The calculation for the resource cost on nodes n and link l_{nm} is given by

$$\psi_n^{\mathrm{node}} = \sum_{f \in \mathcal{F}} \sum_{v_f \in V_s} x_{n,v_f} (\varphi_n^{\mathrm{com}} \cdot R_{v_f}^{\mathrm{com}} + \varphi_n^{\mathrm{mem}} \cdot R_{v_f}^{\mathrm{mem}}), \tag{4}$$

and the calculation for the resource cost on link l_{nm} is given by

$$\psi_{nm}^{\mathrm{link}} = \sum_{f \in \mathcal{F}} \sum_{l_{u_f v_f} \in L_f} \varphi_{mn}^{\mathrm{bw}} \cdot R_{l_{u_f v_f}}^{\mathrm{com}} \cdot x_{nm,u_f v_f}, \tag{5}$$

where x_{n,v_f} is a binary variable, $x_{n,v_f} = 1$ indicates that the v-th VNF of the service request f is successfully embedded at the node n of the physical network. Conversely, if it is unsuccessful, $x_{n,v_f} = 0$. Likewise, the binary variable $x_{nm,u_f v_f}$ represents the deployment status of the virtual link $l_{u_f v_f}$ over the physical link l_{nm}, with $x_{nm,u_f v_f} = 1$ confirming deployment and $x_{nm,u_f v_f} = 0$ signifying its absence. φ_n^{com}, φ_n^{mem} and φ_n^{bw} are the unit prices of computing, memory and bandwidth resources respectively. They change dynamically with resource utilization, and the calculation formulas are given by

$$\begin{aligned}
\varphi_n^{\mathrm{com}} &= (\varphi_{\max}^{\mathrm{com}} - \varphi_{\mathrm{idel}}^{\mathrm{com}})\, \gamma_n^{\mathrm{com}} + \varphi_{\mathrm{idle}}^{\mathrm{com}};\\
\varphi_n^{\mathrm{mem}} &= (\varphi_{\max}^{\mathrm{mem}} - \varphi_{\mathrm{idel}}^{\mathrm{mem}})\, \gamma_n^{\mathrm{mem}} + \varphi_{\mathrm{idle}}^{\mathrm{mem}};\\
\varphi_{nm}^{\mathrm{bw}} &= \left(\varphi_{\max}^{\mathrm{bw}} - \varphi_{\mathrm{idel}}^{\mathrm{bw}}\right) \gamma_n^{\mathrm{bw}} + \varphi_{\mathrm{idle}}^{\mathrm{bw}},
\end{aligned} \tag{6}$$

where $\varphi_{\max}$ and φ_{idle} are the VNF spend when the resource consumption peaks and the VNF spend when the VNF is idle, respectively.

Thus the total resource cost Φ in the network is given by

$$\Phi = \sum_{n \in N} \psi_n^{\mathrm{node}} + \sum_{l_{nm} \in L} \psi_{nm}^{\mathrm{link}}. \tag{7}$$

The total cost U^{total} is the sum of deployment cost and resource cost, expressed as follows:

$$U^{\mathrm{total}} = M + \Phi. \tag{8}$$

To guarantee the user's QoS, NFC orchestration needs to be implemented within SR delay requirements. Delay includes two parts, processing delay of VNF on the physical node n, defined as D_n^{node}, and propagation delay over the physical link l_{nm}, defined as D_{nm}^{link}. Thus the total latency of deploying the request f is given by

$$D_f = \sum_{n \in N} \sum_{v_f \in V_f} x_{n,v_f} D_n^{\mathrm{node}} + \sum_{l_{nm} \in L} \sum_{l_{u_f, f_v} \in L_f} x_{nm,u_f v_f} D_{nm}^{\mathrm{link}}. \tag{9}$$

The total delay D^{total} of all requests can be modeled as:

$$D^{\mathrm{total}} = \sum_{f \in \mathcal{F}} D_f. \tag{10}$$

We aim to minimize the total network cost and delay. Also, the feasible NFC orchestration should satisfy constraints such as resource constraints and delay, so the optimization objective and constraints are given by

$$\min_{M,\Phi,D_f} \alpha U^{\mathrm{total}} + \beta D^{\mathrm{total}} \tag{11a}$$

$$s.t.\ D_f \le T_f, \forall f \in \mathcal{F} \tag{11b}$$

$$\sum_{n \in N} x_{n,v_f} = 1, \forall f \in \mathcal{F} \tag{11c}$$

$$\sum_{f \in \mathcal{F}} \sum_{v_f \in V_f} R_{v_f}^{\mathrm{com}} x_{n,v_f} \le C_n^{\mathrm{com}}, \forall n \in N \tag{11d}$$

$$\sum_{f \in \mathcal{F}} \sum_{v_f \in V_f} R_{v_f}^{\mathrm{mem}} x_{n,v_f} \le C_n^{\mathrm{mem}}, \forall n \in N \tag{11e}$$

$$\sum_{f \in \mathcal{F}} \sum_{l_{u_f v_f} \in L_f} R_{l_{u_f v_f}}^{\mathrm{bw}} x_{nm,u_f v_f} \le C_{nm}^{\mathrm{bw}}, \forall l_{nm} \in L, \tag{11f}$$

where α and β serve as adjustable optimization weights. A higher α steers the policy towards cost-effective strategies, while a higher β favors strategies that reduce overall latency. Constraint (11b) mandates that all successfully deployed VNFs must meet latency QoS requirements, the total processing delay cannot be greater than the maximum tolerated delay. Constraint (11c) guarantees that each VNF within an NFC is deployed on a unique VNF instance, preventing service provision by multiple instances. Constraints (11d) and (11e) set limitations on the resource capacity of network nodes, whereas constraint (11f) addresses the resource constraints on links.

IV. Design Of ANFCOS

This section demonstrates an NFC orchestration strategy based on DRL. We transform the optimization problem into a MDP and extend the action space. The solution uses the MARL method with attention mechanism. In order to enhance the exploration ability of the algorithm, the entropy term is added to optimize the policy gradient to learn the soft value function, and finally the convergence of the method is verified.

A. Problem Transformation

The objective optimization problem needs to be transformed into a proposed solution. In the Markov model, there is a limited set of user requests set $\mathcal{I}$, and each user is defined as $i \in \mathcal{I} = 1, 2, ..., I$; state space S, as the possible environment for all users; a group of actions $a_1, a_2, \ldots a_I$ and observations $o_1, o_2, \ldots o_I$. Using Markov properties, each user i will use the transition probability $P\left(s_i^{'} | s_i, \Pi_i\right)$ to choose an action, where P is the probability that the next state is in $s_i^{'}$ given the current state s_i and the current action space Π_i. Each agent expects the maximum total return value $R_i = \sum_{\mathrm{t=0}}^{\mathrm{T}} \gamma^t r_i^t$, Where γ represents the discount factor and T represents the time range. Specifically, in our scenario, we define the three components of the Markov game model as follows.

1) State: In the t-th time, we define the state $s_i \in \mathcal{S}$ as the remaining resources and VNF deployment situation, which can be described as

$$s\left(t\right) = \left(\mathbb{C}^{\mathrm{com}}, \mathbb{C}^{\mathrm{mem}}, \mathbb{C}^{\mathrm{bw}}, \mathbb{P}\right), \tag{12}$$

where $\mathbb{C}^{\mathrm{com}}$, $\mathbb{C}^{\mathrm{mem}}$, $\mathbb{C}^{\mathrm{bw}}$ represent the sets of remaining computing, memory, and bandwidth resources in the network, respectively. $\mathbb{P}$ denotes the set of VNF types deployed on each device.

2)Action: Actions are defined as node selection $a_{i,n}^{node}$ and link selection $a_{i,nm}^{link}$, that is, allocating server node resources to the VNF and selecting optimal path for connection. As mentioned above, the set of server nodes in the network topology is N and the set of links is L. The action space of the t-th time slot is defined as:

$$\begin{aligned} a\left(t\right) =& (\left\{ a_{i,n}^{node} \,|(i \in \mathcal{I}) \wedge (n \in N) \right\}, \\ & \left\{ a_{i,nm}^{link} \,|(i \in \mathcal{I}) \wedge (l_{nm} \in L) \right\}). \end{aligned} \tag{13}$$

3)Reward: For agents, the reward generated by each action consists of both penalties and contributions towards a common goal. The joint value of rewards is based on the formula (11a). The penalty component of the reward targets that fail to meet certain constraints, specifically, those outlined in (11d), (11e), and (11f). Assuming these constraints share equal priority, we define the combined reward value and its associated penalties accordingly.

$$\left\{ \begin{array}{c} r^o = w^o / (\alpha U^{\mathrm{total}} + \beta D^{\mathrm{total}}) \\ r^p = w^p \left(\chi\left(11d\right) + \chi\left(11e\right) + \chi\left(11f\right)\right) \end{array} \right. , \tag{14}$$

where w^o is the reward factor, $w^o > 0$; w^p is the penalty factor, and $w^p < 0$; $\chi(x) = 1$ if x is false, otherwise $\chi(x) = 0$. So the reward is given by

$$r\left(t\right) = r^o\left(t\right) + r^p\left(t\right). \tag{15}$$

B. Action Strategies

In ANFCOS, operations must adhere to the objectives and rules of NFC orchestration, where RL plays a pivotal role. RL is responsible for determining on which server nodes the VNFs should be deployed and performing detailed evaluations for each VNF deployment. Additionally, link selection optimizes paths and bandwidth to identify the best connection between nodes. This approach, by integrating node selection and link optimization, significantly enhances the efficiency of network resource utilization and deployment flexibility, achieving an optimal balance between network performance and cost-effectiveness.

1) For node selection, our focus is on scenarios involving VNF instance sharing. Once an agent selects a node, it must assess the necessity of deploying a VNF and accordingly allocate resources. This approach enhances resource utilization by avoiding unnecessary VNF deployments, thereby improving overall network efficiency and service performance. Specifically, if the selected node already possesses a VNF type that matches the current request, only resource allocation is required without the need for additional VNF deployment. Conversely, if the selected node lacks the required VNF type, the agent must first deploy the VNF before embedding the request and allocating resources. Hence, the action on the node is given by

$$a_{i,n}^{\mathrm{node}}\left(t\right) = \left\{ \begin{array}{cc} \left[a_{i,n,p}^{\mathrm{deploy}}, a_{i,n,v}^{\mathrm{embed}}\right] & if\ y_{n,p} = 0 \\ \left[a_{i,n,v}^{\mathrm{embed}}\right] & if\ y_{n,p} = 1 \end{array} \right. , \tag{16}$$

where $a_{i,n,p}^{\mathrm{deploy}}$ indicates the action of deploying VNF of type p on device n. $a_{i,n}^{\mathrm{embed}}$ indicates embedding the current VNF v on device n and allocating resources.

2) For link selection, the two primary considerations are the distance between links and the bandwidth utilization. The distance between links, which refers to the number of hops between nodes, significantly affects transmission delay, while bandwidth utilization determines the quality of transmission. Therefore, accurately assessing these two factors is critical for optimizing network performance. The formula to calculate the distance between links, denoted as $R_{l_{n,k,m}}$, is defined as follows:

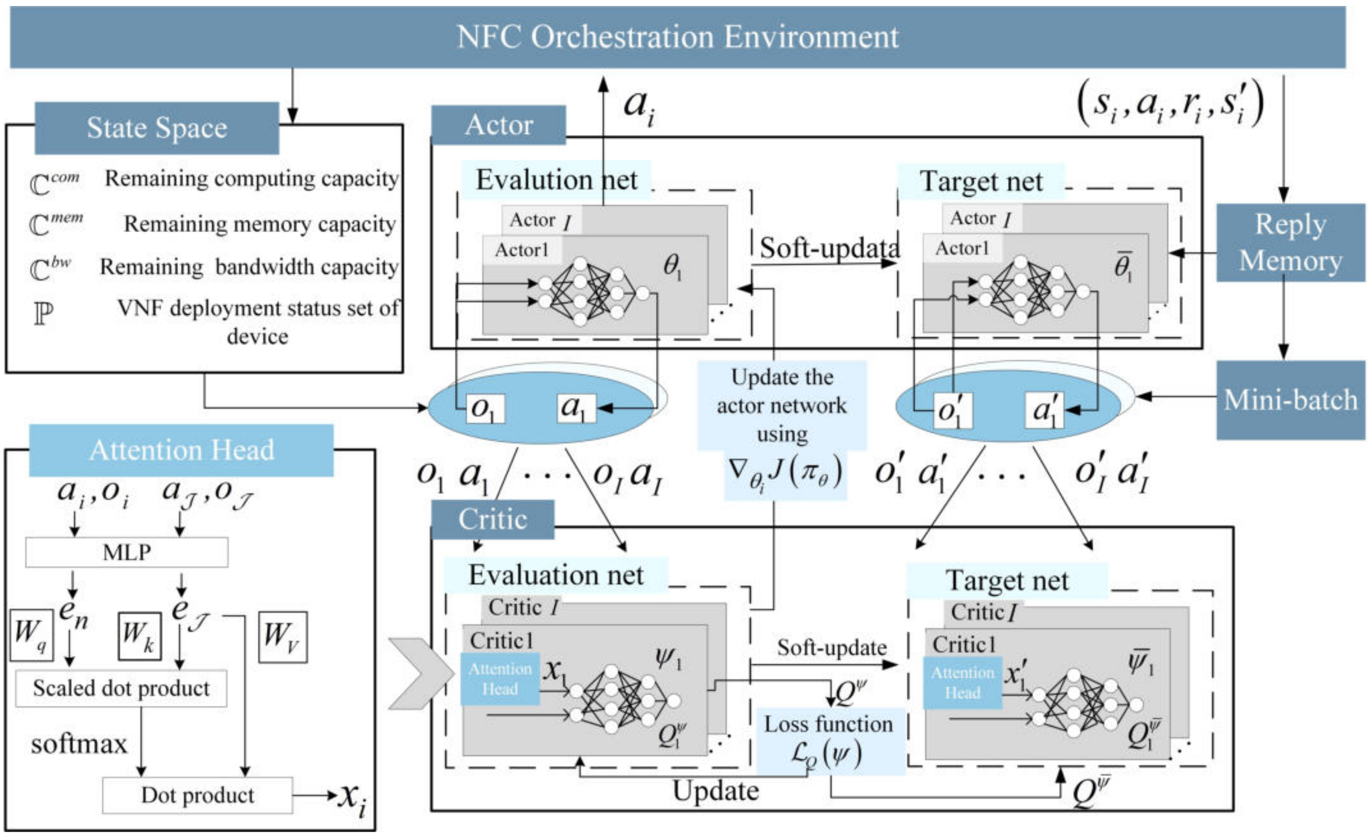


Fig. 2: The ANFCOS algorithm consists of RL and environment components. NFCs from different users act as agents interacting with the intelligent entities, where each entity is equipped with its own critic and actor networks.

$$R_{l_{n,k,m}} = \frac{hop\,(n,k,m)}{hop\,(n,m)}, \tag{17}$$

where $hop(n,m)$ represents the shortest path distance between nodes n and m. Similarly, $hop(n,k,m)$ defines the shortest path through intermediary link k. Regarding bandwidth utilization, in order to prevent path congestion and achieve traffic load balancing, the evaluation of the bandwidth utilization of each link $B_{l_{n,k,m}}$ is given by

$$B_{l_{n,k,m}} = \sum_{k \in l_{nm}} \left(1 - \frac{R_i^{\text{bw}}}{C_k^{\text{bw}} - \gamma_k^{\text{bw}} C_k^{\text{bw}}}\right). \tag{18}$$

as mentioned before, $R_n^{bw}$is the bandwidth request for agent i, C_k^{bw} and γ_k^{bw}respectively is the loan capacity and bandwidth utilization of the link k. Therefore, the trend of each link selection can be defined as:

$$I_{l_{n,k,m}} = \max(B_{l_{n,k,m}}/R_{l_{n,k,m}}), \tag{19a}$$

$$s.t.\ D_i \le T_i, \tag{19b}$$

the link selection process aims to identify a link that not only complies with the delay constraint $D_i <= T_i$ but also maximizes the interest metric $I_{l_{n,k,m}}$. Thus, $a_{i,nm}^{\text{link}} = l_{n,k,m}$. The rules governing the selection of nodes and links for actions are outlined in Algorithm 1.

C. ANFCOS Algorithm

The proposed ANFCOS method is based on the MAAC algorithm in MADRL. This algorithm adopts a centralized training and decentralized execution framework. The “critic” component aggregates state information and actions from all agents and uses an attention mechanism to highlight task-relevant key information, enabling a more accurate evaluation of the agents’ action values. These action values are fed back into the actor network to optimize its decision-making. The “actor” component updates network parameters based on the critic network and outputs strategies using the information observed by the agent. To promote exploration and avoid suboptimal deterministic policies, the method incorporates an entropy term into the policy gradient, refining the learning of the soft value function. The detailed process is shown in Fig. 2.

1) Attention mechanism: The attention mechanism functions in a manner similar to the differentiable key-value memory model. Each agent intuitively queries the other agents for information about their observations and actions and incorporates that information into the estimation of its value function. To calculate the Q-value function $Q_i^{\psi}(o,a)$ for agent i, $i \in \mathcal{I}$, the critic receives observations $o = (o_1, \ldots, o_I)$ and actions $a = (a_1, \ldots, a_I)$ from all agents. The function $Q_i^{\psi}(o,a)$ is a function of agent i’s observation and action, as well as other agents’ contributions:

$$Q_i^{\psi}(o,a) = f_i\left(e_i\left(o_i, a_i\right), x_i\right), \tag{20}$$

Algorithm 1: Action of deployment and embedding rules

Input the $(t-1)$-th VNF deployment action $a_{i,n}(t-1)$ of all agents ;
for *each agent* i **do**
  if $y_{n,q}=1$ **then**
    $a_{i,n}^{\text{node}}(t)=\left[a_{i,n,v}^{\text{embed}}\right]$, the vth VNF of agent i is directly embedded in node n and allocates resources;
  else
    $a_{i,n}^{\text{node}}(t)=\left[a_{i,n,q}^{\text{deploy}},a_{i,n,v}^{\text{embed}}\right]$, Agent n first deploys type q VNF on node n and then allocates resources to embed VNF requests;
  end
  if $a_{i,n,(v-1)}^{\text{embed}}=a_{i,n,v}^{\text{embed}}$ **then**
    break;
  else
    Calculate $R_{l_{n,k,m}}$, $B_{l_{n,k,m}}$ and $I_{l_{n,k,m}}$;
    Select link which $a_{i,nm}^{\text{link}}=l_{n,k,m}$ satisfy max $(I_{l_{n,k,m}})$ and delay requirement $D_i<=T_i$;
  end
end

where f_i is a two-layer multi-layer perceptron (MLP), while e_i is a joint vector of strategy and network environment (2), with $e_i=g_i(a_{v_i},o_{v_i})$ and g_i is embedding function. The contribution from the other agents x_i is a weighted sum of each agent's value:

$$x_i=\sum_{(j\in\mathcal{I})\wedge(j\neq i)}Att_{i,j}, \tag{21}$$

where, the attention function $Att_{i,j}$ is as described in formula (2).

2) Actor network: The set of policies for all agents is denoted as $\pi=\{\pi_1,\ldots,\pi_I\}$, and the parameters of the actor network represents as $\theta=\{\theta_1,\ldots,\theta_I\}$. For each agent i, the actor network $\mu_i(o_i;\theta_i)$ is trained to optimize its parameters θ_i by maximizing the score provided by its corresponding critic network. The actor networks optimize their parameters θ through policy gradient ascent to maximize the cumulative rewards under the policy π. The gradient is computed as:

$$\begin{aligned}\nabla J_{\theta_i}(\pi_\theta)=\mathbb{E}_{o\sim B,a\sim\pi}&\left[\nabla_{\theta_n}\log(\pi_\theta(a_i|o_i))\right.\\&\left.(-\alpha\log(\pi_\theta(a_i|o_i))+A_i(o,a)\right],\end{aligned} \tag{22}$$

where α is an entropy coefficient promoting exploration and preventing premature convergence. At each iteration, experiences $\left\langle s^b,a^b,r^b,s'^b\right\rangle$ are sampled from the replay buffer, and the Q-value feedback from the critic networks is used to refine the parameters until convergence. After the parameters are updated, the parameters guide the policy generation for subsequent decision-making. For a given next state state $s'^b=\left[o'^b_1,\ldots,o'^b_I\right]$, each actor network produces a policy for the next moment based on its respective next observation o'^b_I and updated parameters $\bar{\theta}_i$:

Algorithm 2: ANFCOS algorithm in IIS

Initialize E parallel environments with agents $\mathcal{I}$, actor network parameters θ, evaluation network ψ, target network parameters $\bar{\theta}$ and $\bar{\psi}$, experience replay buffer B;
$T_{update}\leftarrow 0$
for $episode=1\rightarrow\mathcal{M}$ **do**
  Initialize network environment and NFC state;
  for *each step t* **do**
    Each agent i, from its local observation o_i, select action $a_i\sim\pi_i(\cdot|o_i)$;
    Execute actions a_i, get reward r_i and new observation o'_i, store transitions for all environments in B;
    $T_{update}=T_{update}+E$
    if $T_{update}>$ *min step per updates* **then**
      for *each agent n* **do**
        Sample minibatch b in B;
        And update the critic network:
        Calculate $Q_i^{\psi}(o^b_{1\ldots I},a^b_{1\ldots I})$ for all n in parallel;
        Calculate $Q_i^{\bar{\psi}}(o'^b_{1\ldots I},a'^b_{1\ldots I})$ for all n in parallel using target critic;
        Update critic using $L_Q(\psi)$
      end
      for *each agent n* **do**
        Sample $m\times(o_{1\ldots I})\sim B$;
        And update policy network:
        Calculate $Q_i^{\psi}(o^b_{1\ldots I},a^b_{1\ldots I})$ for all n in parallel;
        Update policies using $\nabla_{\theta_i}J(\pi_\theta)$
      end
      Update target parameters by (25) and (26)
      $T_{update}\leftarrow 0$
    end
  end
end

$$\begin{gathered}\pi_{\bar{\theta}_1}(a'_1|o'_1)=\mu_1\left(o'^b_1;\bar{\theta}_1\right)\\\vdots\\\pi_{\bar{\theta}_I}(a'_I|o'_I)=\mu_I\left(o'^b_I;\bar{\theta}_I\right),\end{gathered} \tag{23}$$

where a' and o' refer to the actions and observations, respectively, for the next moment.

3) Critic networks: The critic networks optimize their parameters $\psi=\{\psi_1,\ldots,\psi_I\}$ to accurately estimate the action-value $Q_i^{\psi}(s,a)$ for the policies generated by the actor networks. The parameters are updated using the temporal difference (TD) method, which minimizes the TD error between the predicted Q-value and the target Q-value. This is achieved by applying the gradient descent with the following loss function:

$$\mathcal{L}_Q(\psi)=\sum_{i=1}^{I}\mathbb{E}_{(o,a,r,o')\sim B}\left[\left(Q_i^{\psi}(o,a)-y_i\right)^2\right], \tag{24}$$

where $y_i = r_i + \gamma \mathbb{E}_{a' \sim \pi_{\bar{\theta}(o')}} \left[Q_i^{\bar{\psi}}(o', a') - \alpha log(\pi_{\bar{\theta}}(a_i'|o_i')) \right]$, formulate the target action value function.

4) Target actor and critic networks update: Update the target critic $Q_{1 \dots I}^{\bar{\psi}}(o, a)$ parameters using equation (25) and target actor $\pi_{\theta_{1 \dots \mathcal{N}}}(a_n | o_n)$ with equation (26), using the update rate τ

$$\bar{\psi} = \bar{\psi} * (1 - \tau) + \psi * \tau, \tag{25}$$

$$\bar{\theta} = \bar{\theta} * (1 - \tau) + \theta * \tau. \tag{26}$$

5) Advantage Function: As shown in [37], an advantage function using a baseline that only marginalizes out the actions of the given agent from $Q_i^{\psi}(o, a)$, can help solve the multi-agent credit assignment problem. In other words, by comparing the value of a specific action to the value of the average action for the agent, with all other agents fixed, we can learn whether said action will cause an increase in expected return or whether any increase in reward is attributed to the actions of other agents. The form of this advantage function is shown below:

$$A_i(o, a) = Q_i^{\psi}(o, a) - b(o, a_{\mathcal{J}}), \tag{27}$$

where $\mathcal{J}$ indicate all agent set except agent i and $b(o, a_{\mathcal{J}})$ calculation formula shown below:

$$b(o, a_{\mathcal{J}}) = \mathbb{E}_{a_i \sim \pi_i(o_i)} \left[Q_i^{\psi}(o, (a_i, a_{\mathcal{J}})) \right]. \tag{28}$$

D. Convergence Proof

In this paper, to encourage exploration and avoid convergence to non-optimal deterministic policies, ANFCOS learns a soft value function by modifying the policy gradient to incorporate an entropy term. Soft value function usually contains soft policy evaluation and soft policy improvement [38]. The traditional approach uses Bellman's equation to implement policy evaluation, updating the Q-value until convergence, and the formula for updating the Q-value is given by

$$\mathrm{Q}_{\pi} = r(o, a) + \lambda E_{o' \sim B, a' \sim \pi} Q_{\pi}(o', a'), \tag{29}$$

and the policy improvement is given by

$$\pi^{\text{new}} = \arg\max_a Q_{\pi^{\text{old}}}(o, a). \tag{30}$$

In soft policy iteration, soft policy evaluation updates the Q-value function through soft Bellman backup. Therefore the improved Q-value update formula in this paper is given by

$$Q_{\pi} = r(o, a) + \gamma \mathbb{E}_{o' \sim B, a' \sim \pi} \left[Q_{\pi}(o', a') - \alpha \log(\pi(a' | o')) \right], \tag{31}$$

and soft policy improvement is given by

$$\begin{aligned} \pi^{\text{new}} = & \arg\min_{\pi' \in \Pi} D_{KL} \\ & \left(\pi'(a|o) \middle\| \exp(Q_{\pi^{\text{old}}}(o, a)) - \log Z^{\pi^{\text{old}}}(o) \right). \end{aligned} \tag{32}$$

where Π is the set of optional actions and the partition function $Z^{\pi^{\text{old}}}$ normalizes the policy distribution. The Q value in RL represents the expected reward that can be obtained by taking an action in a given state. Thus, RL convergence should be expressed as repeated applications of soft policy evaluation and soft policy improvement from any policy $\pi \in \Pi$ will converge to the policy π^* and for all $\pi \in \Pi$ there is $Q_{\pi^*}(o, a) \geq Q_{\pi}(o, a)$. Thus, we need to prove that: 1) the soft Q-values will eventually converge after numerous iterations; and 2) when updating the policy π, $Q_{\pi^{\text{old}}} \leq Q_{\pi^{\text{new}}}$.

1) Prove the convergence of Q-values: We first simplify the soft Q-value update formula presented as equation 31:

$$\begin{aligned} Q_{\pi} &= r(o, a) + \gamma \mathbb{E}_{o' \sim B, a' \sim \pi} \left[Q_{\pi}(o', a') - \alpha \log(\pi(a' | o')) \right] \\ &= r(o, a) + \gamma \mathbb{E}_{o' \sim B} \\ &\quad \left[\mathbb{E}_{a' \sim \pi} \left[Q_{\pi}(o', a') - \alpha \log(\pi(a_n' | o_n')) \right] \right] \\ &= r(o, a) + \gamma \mathbb{E}_{o' \sim B, a' \sim \pi} \left[Q_{\pi}(o', a') \right] + \\ &\quad \gamma \mathbb{E}_{o' \sim B} \mathbb{E}_{a' \sim \pi} \left[-\alpha \log(\pi(a_n' \mid o_n')) \right] \\ &= r(o, a) + \gamma \alpha \mathbb{E}_{o' \sim B} \mathcal{H}(\pi(\bullet \mid o')) + \\ &\quad \gamma \mathbb{E}_{o' \sim B, a' \sim \pi} \left[Q_{\pi}(o', a') \right], \end{aligned} \tag{33}$$

where $\mathcal{H}(\pi(\bullet \mid o'))$ is the expression for entropy. Define the entropy-augmented reward as: $\mathrm{r}^{\text{soft}}(o, a) = r(o, a) + \alpha \gamma \mathbb{E}_{o' \sim B, a' \sim \pi} \left[\mathcal{H}(\pi(\bullet | o')) \right]$, rewrite the update rule as

$$Q_{\pi} = r^{\text{soft}}(o, a) + \gamma \mathbb{E}_{o' \sim B, a' \sim \pi} \left[Q_{\pi}(o', a') \right]. \tag{34}$$

the equation 34 resembles the traditional Bellman equation 29. By applying the standard results on the convergence of policy evaluation, as referenced in [39], it is proven that the Q values will eventually converge to the soft Q values associated with policy π.

2) Proof of convergence to optimal strategy: We need to prove that the converged strategy π^* is the strategy that maximizes reward. According to the equation 32, policy updates minimize the Kullback-Leibler (KL) divergence between the new policy π^{new} and a distribution influenced by the Q-values from the previous policy π^{old}. In this context, $\exp(Q_{\pi^{\text{old}}}(o, a))$ assigns weights to actions based on their Q-values, resulting in higher Q-values leading to higher weights for the corresponding actions. The minimization of the KL divergence ensures the new policy is not inferior to the previous one due to its mathematical properties. The agents strive to optimize their policy to increase their expected discounted returns, formalized as $J(\pi) = E_{o \sim B, a \sim \pi} \left[-\alpha \log(\pi(a|o)) + Q_{\pi} - b(o, a_{\mathcal{J}}) \right]$. Following this approach, it's guaranteed that $J(\pi^{\text{new}}) \geq J(\pi^{\text{old}})$, hance:

$$\begin{aligned} J(\pi^{\text{new}}) &= E_{o \sim B, a \sim \pi} \left[-\alpha \log(\pi^{\text{new}}(a \mid o)) + Q_{\pi^{\text{old}}} - b(o, a_{\mathcal{J}}) \right] \\ &\geq E_{o \sim B, a \sim \pi} \left[-\alpha \log(\pi^{\text{old}}(a \mid o)) + Q_{\pi^{\text{old}}} - b(o, a_{\mathcal{J}}) \right] \\ &= J(\pi^{\text{old}}), \end{aligned} \tag{35}$$

thus, consider the soft Bellman equation:

$$\begin{aligned} Q_{\pi^{\text{old}}} &= r(o, a) + \gamma \mathbb{E}_{o' \sim B, a' \sim \pi} \\ &\qquad \left[Q_{\pi^{\text{old}}}(o', a') - \alpha \log(\pi^{\text{old}}(a' | o')) \right] \\ &\leq r(o, a) + \gamma \mathbb{E}_{o' \sim B, a' \sim \pi} \\ &\qquad \left[Q_{\pi^{\text{old}}}(o', a') - \alpha \log(\pi^{\text{new}}(a' | o')) \right] \\ &\quad \vdots \\ &\leq Q_{\pi^{\text{new}}}(o, a), \end{aligned} \tag{36}$$

let π_i be the policy at iteration i. It follows from the above proof that the sequence Q_{π_i} is monotonically increasing and convergent, thus the sequence converges to some π^* and π^* is optimal in Π.

V. SIMULATION RESULTS

A. Simulation Setup

To verify the convergence and performance of the ANFCOS method, a series of simulation experiments were conducted with the parameters outlined in Table II. The core feature of 6G IIS is to provide personalized services through diverse smart terminals and their collaboration, ensuring equal access to smart and digital services for all users worldwide. Therefore, the experimental design prioritizes the universality, personalization, and service diversity of 6G IISs. The experiments were based on a complex physical topology with 50 nodes, where the probability of inter-node connections was 35%. SR involved eight different types of VNFs, randomly combined to form various SR types, each comprising between two and six different types of VNFs.

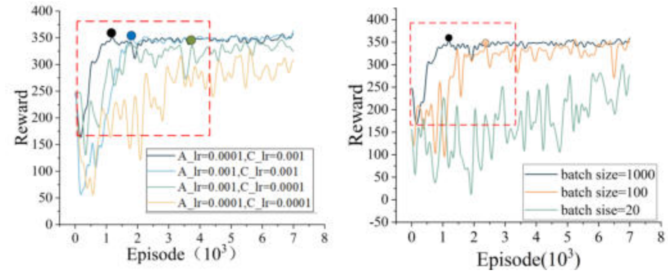
TABLE II: SIMULATION PARAMETERS

Symbol	Value	Description
$\|N\|$	50	Number of devices
$L(\%)$	35%	Connection rate between devices
C_n^{com},C_n^{mem}	$[50, 80]$	Computing and storage resources of the device
C_{nm}^{nm}	$[80, 100]$	Bandwidth resources of physical links
$\|P\|$	8	Number of VNF types
$\|V_f\|$	$[3, 6]$	Number of VNFs comprising the SR f
$\|P_f\|$	$[2, 6]$	Number of VNF types comprising the SR f
T_f	$[80, 100]$	Maximum tolerable delay for SR f
batch-size	1000	Number of training examples in one forward
critic-lr	$1*10^{-2}$	Learning rates of critic network
actor-lr	$1*10^{-3}$	Learning rates of actor-network
critic-hidden-dim	128	Critic network hidden layer dimensions
actor-hidden-dim	128	Actor network hidden layer dimensions
Discount factor	0.9	Weigh the importance of current rewards against future rewards

B. Parameter Impact Analysis

Different learning rates and batch sizes significantly impact the convergence accuracy and speed of ANFCOS. As illustrated in Fig. 3, we assess the performance of ANFCOS under various parameter settings by changing the number of episodes and observing the reward variations.

Figure. 3a analyzes the influence of the learning rate of the critic network C_lr and the actor network A_lr. Setting both networks' learning rates at 0.0001 does not achieve convergence. Increasing the learning rate of the critic network to 0.001 significantly boosts convergence and effectiveness. However, further increasing the actor network's learning rate to 0.001 shows negligible improvement. The optimal configuration is to set the actor network's learning rate to 0.0001 and the critic network's learning rate to 0.001 achieving superior convergence velocity and reward outcomes. This demonstrates



(a) The effect of learning rate on reward.

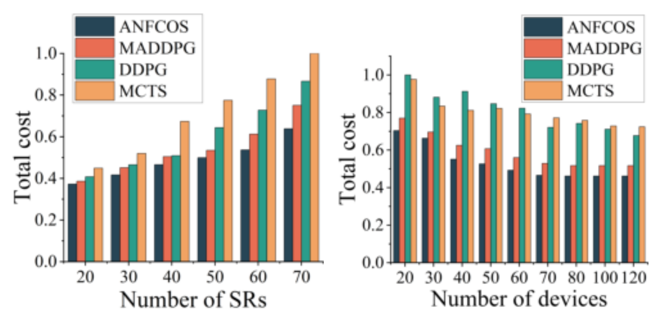
(b) The effect of batch size on reward.

Fig. 3: The impact of different parameter settings on the convergence of the algorithm.

ANFCOS's ability to effectively discern the contributions of individual agents to the overall system rewards, thus facilitating more accurate learning and optimization.

As illustrated in Fig. 3b, the batch size is determined through experimental tuning. An experience replay buffer with size of 1×10^5 and random sampling is configured. When the batch size is small, the algorithm can not convergence. As the batch size increases, the neural network's convergence speed becomes faster and higher rewards increase. When the batch size reaches 1000, the convergence speed still improves. However the computational demand increases significantly, potentially exceeding hardware limitations.

According to subsection IV-D, Fig. 3 shows that ANFCOS can achieve a better convergence effect by expanding action space and introducing a resolvable attention mechanism based on MAAC.



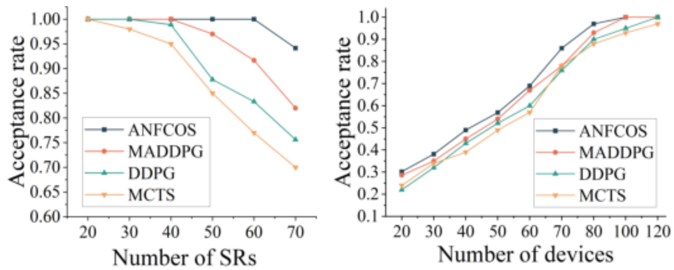
(a) Total cost with different numbers of SRs.

(b) Total cost with different numbers of devices.

Fig. 4: Total cost for different number of SRs and number of devices.

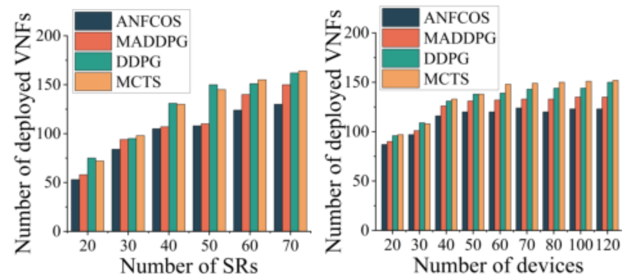
C. Performance Evaluation

To evaluate the performance of ANFCOS, it is compared with DDPG, MADDPG, and MCTS based on acceptance rate, cost, number of VNFs deployed, delay, and network revenue. These algorithms serve as benchmarks for more effective comparison. Where the MADDPG algorithm was used as an ablation method for ANFCOS to assess the impact of removing the attention mechanism.



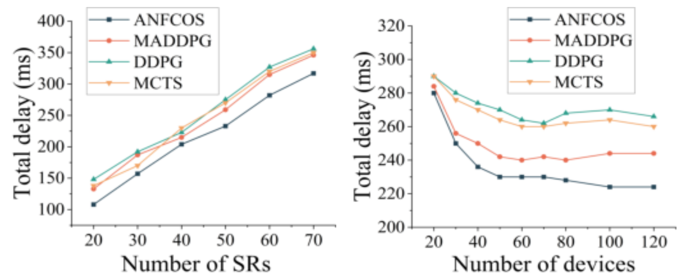
(a) Acceptance rate with different number of SRs.
(b) Acceptance rate with different number of devices.

Fig. 5: Acceptance rate with different number of SRs and number of devices.



(a) Number of VNF deployments with different number of SRs.
(b) Number of VNF deployments with different number of SRs.

Fig. 6: Number of deployed VNFs with different number of SRs and number of devices.



(a) Total delay with the different number of SRs.
(b) Total delay with different numbers of devices.

Fig. 7: Total delay with different number of SRs and number of devices.

a) DDPG-based Approach [29]: DDPG algorithm is a model-free, off-policy actor-critic approach. It uses a deterministic policy gradient that allows for efficient estimation of the optimal policy by directly maximizing the expected return.

b) MADDPG-based Approach [32]: ANFCOS extends MADDPG by adding an attention mechanism to enhance agent interaction modeling. In the ablation experiments, we removed the attention mechanism and compared the performance with the original MADDPG to assess its impact.

c) MCTS-based Approach [28]: MCTS builds a sparse search tree and selects actions through Monte Carlo sampling. The path involving the root, intermediate nodes with the highest average values, and leaf nodes defines the operational strategy for orchestrating NFC.

1) Cost: Figure. 4 presents the cost dynamics of various methods in response to an upsurge in SRs and the physical devices.

Figure. 4a demonstrates that with the rising number of SRs, the deployment of VNFs increases, and the network requires more resources to meet these demands, leading to a gradual increase in costs. In this context, the ANFCOS strategy outperforms other benchmarks in cost optimization. Fig. 4b further analyzes the impact of varying numbers of devices on NFC orchestration, indicating that higher device counts enable the network to decrease resource costs by broadening VNF deployments, thus mitigating overall costs. Remarkably, at a device count of 60, while costs for MCTS and DDPG level off, ANFCOS continues to offer the most cost-effective solution.

Combining Fig. 4a and Fig. 4b, ANFCOS achieves a cost reduction of approximately 8.2% compared to MADDPG, 22.1% compared to DDPG, and 28.6% compared to MCTS, significantly enhancing cost efficiency.

2) Acceptance Rate: Figure. 5 presents the acceptance rate dynamics of various methods in response to an upsurge in service requests and physical devices.

Figure. 5a compares the acceptance rates of different algorithms with the increasing number of SRs. After a certain number of SRs, the acceptance rates generally decline. This is because the resources of the network are limited. As the number of deployed SRs increases, network resources are occupied and cannot serve more SRs. Fig. 5b illustrates the performance of different algorithms in terms of acceptance rate as the number of devices increases. As the number of nodes in the network grows, each algorithm gains more resources to optimize decision-making and execution when handling SR. As a result, the acceptance rate of each algorithm gradually increases.

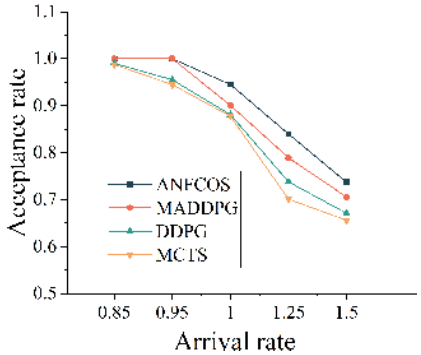
ANFCOS reduces the number of failed SRs that violate constraints through the attention mechanism and enhances reward, resulting in the acceptance rates about 10% higher than MADDPG and 20% higher than DDPG and MCTS.

3) Number of VNFs deployed: Figure. 6 presents the actual number of deployed VNF of various methods in response to an upsurge in service requests and physical devices.

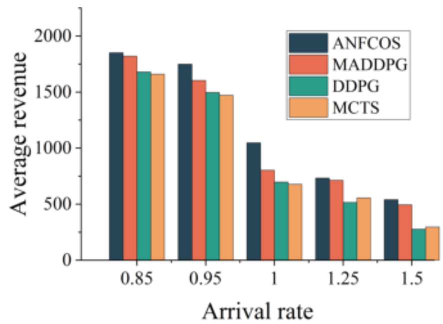
Figure. 6a compares the number of VNFs deployed by different approaches as the SR increments. As the number of service requests grows, there is a need to deploy more VNFs to process these requests and balance network load, the number of VNF deployments has also increased. In Fig. 6b, with an increase in the number of devices, load balancing and path optimization can be achieved by deploying VNFs across additional devices, leading to an increase in the number of VNFs. When the number of devices reaches 80, the quantity of deployed VNFs gradually stabilizes at around 130.

Overall, the actual number of VNFs deployed by ANFCOS is better than the other three methods, and lower number of VNF deployments means optimized deployment costs.

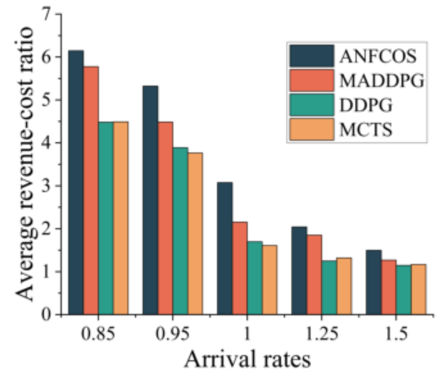
4) Delay: Figure. 7 presents the delay of various methods in response to an upsurge in service requests and physical



(a) Acceptance rate with different arrival rate.



(b) Average revenue with different arrival rates.



(c) Average revenue-cost ratio with different arrival rates.

Fig. 8: The influence of different arrival rates on the orchestration scheme.

devices.

Figure. 7a plots the latency of different approaches as the number of SRs increases. As the number of SRs increases, some of the requests cannot be deployed on the shortest routing path, resulting in a rise in latency. Compared to the benchmarks, ANFCOS performs the best in latency by optimizing the action strategy for link selection. Fig. 7b compares the delay variations of the different approaches, as the number of devices increases under a fixed number of SRs. With the increases in the number of devices and resources available to the network, VNF deployment can be decentralized to optimize the latency. When the number of devices reaches about 70, the latency stabilizes.

Experimental results demonstrate that ANFCOS effectively reduces cross-node communication delay by prioritizing short-distance links and considering both physical distance and bandwidth utilization. ANFCOS reduces the total delay by an average of 7.7% compared to MADDPG, and by 12.7% and 10.8% compared to DDPG and MCTS, respectively.

5) Network Revenue: Figure. 8 simulates the impact of sending service requests to the network at a certain arrival rate in reality on the network acceptance rate and network revenue.

Requests arrive at the network at different arrival rates within 50-time slots. As arrival rates increase, both acceptance rates and average revenue decrease as shown in Fig. 8a. This trend occurs because when the acceptance rate increases, requests arrive more frequently. However, the physical resources are still occupied by the previous NFC and new requests have to be rejected. In Fig. 8b, as the arrival rate continues to increase, the average revenue decreases. This decrease is that the penalty cost increases as the acceptance rate decreases. Consequently, the average revenue-cost ratio, as depicted in Fig. 8c, also experiences a decline. Nonetheless, when analyzing the combined data on average revenue and the average revenue-cost ratio from Figures. 8b and 8c, the effectiveness of our algorithm in optimizing costs under these conditions becomes evident.

Compared with single-agent reinforcement learning DDPG and machine learning MCTS, MARL performance generally shows better performance and robustness. Compared with the MADDPG algorithm, which is also a multi-agent, the acceptance rate and cost of our algorithm under different SR arrival rates are better than MADDPG without an attention mechanism, which shows that the attention mechanism added to ANFCOS can improve the performance of RL performance.

VI. CONCLUSIONS

The adaptive NFC orchestration strategy represents the fundamental technology for implementing advanced and intelligent services in future 6G systems. For 6G IIS, how to improve QoS and reduce operating costs through adaptive NFC orchestration is a difficult point. In this paper, we solve the problems of association orchestration of heterogeneous devices due to differences in applications in 6G IIS, and propose an ANFCOS. This policy connects VNFs to devices and uses a focus mechanism for NFC to address the need for personalized and diversified services in the future. Simulation results show that ANFCOS can effectively reduce operating costs and end-to-end latency while maintaining user QoS.

Future work aims to refine ANFCOS to accommodate a wider range of 6G application scenarios. Overall, the adaptive NFC orchestration strategy represents a foundational technology for enabling IIS in future 6G systems. Its adaptability and optimization capabilities hold the potential to revolutionize the way we deliver and consume intelligent services in the 6G era.

REFERENCES

[1] "6G typical scenarios and key capabilities white paper," tech. rep., IMT-3020 (6G) Promotion Group, 2021. http://www.caict.ac.cn/kxyj/qwfb/ztbg//202207/P020220724437616473703.pdf.

[2] H. Yang, W. Liu, X. Wang, *et al.*, "Group sparse space information network with joint virtual network function deployment and maximum flow routing strategy," *IEEE Transactions on Wireless Communications*, vol. 22, no. 8, pp. 5291–5305, 2023.

[3] J. Li, R. Wang, and K. Wang, "Service function chaining in industrial internet of things with edge intelligence: A natural actor-critic approach," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 491–502, 2023.

[4] S. Yang, F. Li, R. Yahyapour, *et al.*, "Delay-sensitive and availability-aware virtual network function scheduling for NFV," *IEEE Transactions on Services Computing*, vol. 15, no. 1, pp. 188–201, 2022.

[5] O. Houidi, O. Soualah, I. Houidi, *et al.*, "Energy efficient VNF-FG embedding via attention-based deep reinforcement learning," in *2023 19th International Conference on Network and Service Management (CNSM)*, pp. 1–7, IEEE, 2023.

[6] M. Dolati, S. H. Rastegar, A. Khonsari, *et al.*, "Layer-aware containerized service orchestration in edge networks," *IEEE Transactions on Network and Service Management*, vol. 20, no. 2, pp. 1830–1846, 2023.

[7] T. V. Doan, G. T. Nguyen, M. Reisslein, *et al.*, "SAP: Subchain-aware NFV service placement in mobile edge cloud," *IEEE Transactions on Network and Service Management*, vol. 20, no. 1, pp. 319–341, 2023.

[8] J. Zhang, Y. Liu, Z. Li, *et al.*, "Forecast-assisted service function chain dynamic deployment for SDN/NFV-enabled cloud management systems," *IEEE Systems Journal*, vol. 17, no. 3, pp. 4371–4382, 2023.

[9] B. Yi, X. Wang, and M. Huang, "A generalized VNF sharing approach for service scheduling," *IEEE Communications Letters*, vol. 22, no. 1, pp. 73–76, 2018.

[10] N. Tastevin, M. Obadia, and M. Bouet, "A graph approach to placement of service functions chains," in *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pp. 134–141, 2017.

[11] C. Ren, H. Li, Y. Li, *et al.*, "On efficient service function chaining in hybrid software defined networks," *IEEE Transactions on Network and Service Management*, vol. 19, no. 2, pp. 1614–1628, 2022.

[12] Z. Yuan, L. Luo, D. Guo, *et al.*, "To deploy new or to deploy more?: An online SFC deployment scheme at network edge," *IEEE Internet of Things Journal*, vol. 11, no. 2, pp. 2336–2350, 2024.

[13] A. Dalgkitsis, L. A. Garrido, F. Rezazadeh, *et al.*, "SCHE2MA: Scalable, energy-aware, multidomain orchestration for beyond-5g URLLC services," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 7, pp. 7653–7663, 2023.

[14] R. Solozabal, J. Ceberio, A. Sanchoyerto, L. Zabala, B. Blanco, and F. Liberal, "Virtual network function placement optimization with deep reinforcement learning," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 2, pp. 292–303, 2020.

[15] Y. Yue, B. Cheng, M. Wang, *et al.*, "Throughput optimization and delay guarantee VNF placement for mapping SFC requests in NFV-enabled networks," *IEEE Transactions on Network and Service Management*, vol. 18, no. 4, pp. 4247–4262, 2021.

[16] R. Borrell, G. Oyarzun, D. Dosimont, and G. Houzeaux, "Parallel sfc-based mesh partitioning and load balancing," in *2019 IEEE/ACM 10th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems (ScalA)*, pp. 72–78, 2019.

[17] Y. Liu, C. Zhang, H. Yang, S. Zhang, X. Wang, and F. Li, "Deep reinforcement learning based reliability aware sfc placement in multi-domain networks," in *2023 15th International Conference on Communication Software and Networks (ICCSN)*, pp. 215–219, 2023.

[18] H. Chen, S. Wang, G. Li, *et al.*, "Distributed orchestration of service function chains for edge intelligence in the industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 9, pp. 6244–6254, 2021.

[19] S. Jeon, H. Lee, V. K. Kaliappan, *et al.*, "Multiagent reinforcement learning based on fusion-multiactor-attention-critic for multiple-unmanned-aerial-vehicle navigation control," *Energies*, vol. 15, no. 19, 2022.

[20] N. He, S. Yang, F. Li, *et al.*, "A-DDPG: Attention mechanism-based deep reinforcement learning for NFV," in *2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS)*, pp. 1–10, IEEE, 2021.

[21] S. Iqbal and F. Sha, "Actor-attention-critic for multi-agent reinforcement learning," in *International conference on machine learning*, pp. 2961–2970, PMLR, 2019.

[22] S. Wang, C. Yuen, W. Ni, *et al.*, "Multiagent deep reinforcement learning for cost-and delay-sensitive virtual network function placement and routing," *IEEE Transactions on Communications*, vol. 70, no. 8, pp. 5208–5224, 2022.

[23] J. Xu, X. Cao, Q. Duan, *et al.*, "Service function chain deployment using deep Q-learning and tidal mechanism," *IEEE Internet of Things Journal*, 2023.

[24] H. Cao, J. Du, H. Zhao, *et al.*, "Resource-ability assisted service function chain embedding and scheduling for 6G networks with virtualization," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 4, pp. 3846–3859, 2021.

[25] F. Bari, S. R. Chowdhury, R. Ahmed, *et al.*, "Orchestrating virtualized network functions," *IEEE Transactions on Network and Service Management*, vol. 13, no. 4, pp. 725–739, 2016.

[26] S. Xu, Y. Li, S. Guo, *et al.*, "Cloud–edge collaborative SFC mapping for industrial IoT using deep reinforcement learning," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 6, pp. 4158–4168, 2021.

[27] X. Wang, S. Wang, X. Liang, *et al.*, "Deep reinforcement learning: a survey," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[28] S. Haeri and L. Trajković, "Virtual network embedding via monte carlo tree search," *IEEE transactions on cybernetics*, vol. 48, no. 2, pp. 510–521, 2017.

[29] L. Gu, D. Zeng, W. Li, *et al.*, "Deep reinforcement learning based VNF management in geo-distributed edge computing," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 934–943, IEEE, 2019.

[30] Z. Zhang, Y.-S. Ong, D. Wang, *et al.*, "A collaborative multiagent reinforcement learning method based on policy gradient potential," *IEEE transactions on cybernetics*, vol. 51, no. 2, pp. 1015–1027, 2019.

[31] H. Xuan, Y. Zhou, X. Zhao, *et al.*, "Multi-agent deep reinforcement learning algorithm with self-adaption division strategy for VNF-sc deployment in SDN/NFV-enabled networks," *Applied Soft Computing*, vol. 138, p. 110189, 2023.

[32] Y. Zhu, H. Yao, T. Mai, *et al.*, "Multiagent reinforcement-learning-aided service function chain deployment for internet of things," *IEEE Internet of Things Journal*, vol. 9, no. 17, pp. 15674–15684, 2022.

[33] P. Parnika, R. B. Diddigi, S. K. R. Danda, *et al.*, "Attention actor-critic algorithm for multi-agent constrained co-operative reinforcement learning," *arXiv preprint arXiv:2101.02349*, 2021.

[34] D. Chen, Q. Qi, Q. Fu, J. Wang, J. Liao, and Z. Han, "Transformer-based reinforcement learning for scalable multi-uav area coverage," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 8, pp. 10062–10077, 2024.

[35] J. Xue, K. Yu, T. Zhang, H. Zhou, L. Zhao, and X. Shen, "Cooperative deep reinforcement learning enabled power allocation for packet duplication urllc in multi-connectivity vehicular networks," *IEEE Transactions on Mobile Computing*, vol. 23, no. 8, pp. 8143–8157, 2024.

[36] K. Li, X. Wang, Q. He, B. Yi, A. Morichetta, and M. Huang, "Cooperative multiagent deep reinforcement learning for computation offloading: A mobile network operator perspective," *IEEE Internet of Things Journal*, vol. 9, no. 23, pp. 24161–24173, 2022.

[37] J. Foerster, G. Farquhar, T. Afouras, *et al.*, "Counterfactual multi-agent policy gradients," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, 2018.

[38] T. Haarnoja, A. Zhou, P. Abbeel, *et al.*, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80 of *Proceedings of Machine Learning Research*, pp. 1861–1870, PMLR, 10–15 Jul 2018.

[39] R. S. Sutton and A. G. Barto, "Reinforcement learning: An introduction," *Robotica*, vol. 17, no. 2, pp. 229–235, 1999.

Yi Wen (wenyiiii@foxmail.com) received the B.S degree from Guizhou University, China, in 2022 and is currently pursuing a master's degree in the College of Big Data and Information Engineering, Guizhou University. Her main research interests are orchestration and scheduling of network function chains, and reinforcement learning.

Sai Zou (dr-zousai@foxmail.com) received a Ph.D. degree in Communication Engineering from Xiamen University, China, in 2018. He received an M.S. degree in Software Engineering from Hunan University, China, in 2007. He received a B.S. degree in Computer Science from Hengyang Normal University, China, in 2004.

Currently, he is the director of Guizhou Cloud-Network Collaborative Innovation Center and Guizhou Cloud-Network Collaborative Deterministic Transmission Engineering Center and a professor at Guizhou University, China. He was a visiting research scholar at CSIRO, Sydney, Australia, from 2017 to 2018; and was a postdoctoral fellow at UESTC, China, from 2020 to 2022. He won the title of "Thousand Innovative Talents" in Guizhou Province and "Young Bayu Scholar" in Chongqing. His current research interests include 5G networks and network slices and next-generation networks.

Dr Zou serves as a TPC member in IEEE VTC 2020, and a Local Organizing Chair for ICCDA 2023, and a Guest Editor for Mobile Information Systems 2022 and IET Blockchain 2023.

Yanglong Sun received the Ph.D. degree from Xiamen University, China, in 2022, received the B.S. degree and M.S degree from Zhengzhou University, China, in 2011 and 2014 respectively. He is currently an assistant professor at Navigation College, Jimei University. He was a Visiting Ph.D. Student at the Broadband Communications Research (BBCR) Laboratory, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, from 2018 to 2019. His research interests include vehicular ad hoc networks, resource allocation in IoV, mobile 5G networks, and intelligent transportation systems.

Wei Ni (M'09–SM'15–F'24) received the B.E. and Ph.D. degrees in Communication Science and Engineering from Fudan University, Shanghai, China, in 2000 and 2005, respectively. He is a Principal Research Scientist at CSIRO, Sydney, Australia, and an Adjunct Professor at the University of Technology Sydney and an Honorary Professor at Macquarie University. He was a Postdoctoral Research Fellow at Shanghai Jiaotong University from 2005 to 2008; Deputy Project Manager at Bell Labs, Alcatel/Alcatel-Lucent from 2005 to 2008; and Senior Researcher at Devices R&D, Nokia from 2008 to 2009. He has authored five book chapters, over 200 journal papers, over 100 conference papers, 25 patents, and ten standard proposals accepted by IEEE. His research interests include machine learning, cybersecurity, network security, privacy, and their applications to 6G/B6G system efficiency and integrity.

Dr Ni has been an Editor for IEEE Transactions on Vehicular Technology since 2022 and an Editor for IEEE Transactions on Wireless Communications since 2018. He served first as the Secretary, then Vice-Chair and Chair of the IEEE NSW VTS Chapter from 2015 to 2022, Track Chair for VTC-Spring 2017, Track Co-chair for IEEE VTC-Spring 2016, Publication Chair for BodyNet 2015, and Student Travel Grant Chair for WPMC 2014.

Kai Li (S'09–M'14–SM'20) received the B.E. degree from Shandong University, China, in 2009, the M.S. degree from The Hong Kong University of Science and Technology, Hong Kong, in 2010, and the Ph.D. degree in computer science from The University of New South Wales, Sydney, NSW, Australia, in 2014. Currently, he is a Visiting Research Scholar with the School of Electrical Engineering and Computer Science, TU Berlin, Germany, and a Senior Research Scientist with the CISTER Research Centre, Porto, Portugal. He is also a CMU-Portugal Research Fellow, jointly supported by Carnegie Mellon University (CMU), Pittsburgh, PA, USA, and the Foundation for Science and Technology (FCT), Lisbon, Portugal. From 2023 to 2024, he was a Visiting Research Scientist with the Division of Electrical Engineering, Department of Engineering, University of Cambridge, UK. In 2022, he was a Visiting Research Scholar with the CyLab Security and Privacy Institute, CMU. Prior to this, he was a Post-Doctoral Research Fellow with the SUTD-MIT International Design Centre, Singapore University of Technology and Design, Singapore, from 2014 to 2016. He has also held positions as a Visiting Research Assistant with the ICT Centre, CSIRO, Brisbane, QLD, Australia, from 2012 to 2013, and a full-time Research Assistant with the Mobile Technologies Centre, The Chinese University of Hong Kong, Hong Kong, from 2010 to 2011. He has been an Associate Editor of journals, such as *Internet of Things* (Elsevier) since 2024, *Nature Computer Science* (Springer) since 2023, *Computer Communications* (Elsevier) and *Ad Hoc Networks* (Elsevier) since 2021, and IEEE ACCESS from 2018 to 2024.

Hongfeng Gao received his B.S. degree in 1996 from College of physics and Electronic Science, Guizhou University, his M.S. degree in 2007 from College of Computer Science, Guizhou University. Currently he is an associate professor at Guizhou University, Guiyang, China. His research interests include network and information security.