Load-Aware Network Resource Orchestration in LEO Satellite Network: A GAT-Based Approach

Jingchao He, *Student Member*, *IEEE*, Nan Cheng, *Senior Member*, *IEEE*, Zhisheng Yin, *Member*, *IEEE*, Haibo Zhou, *Senior Member*, *IEEE*, Conghao Zhou, *Member*, *IEEE*, Khalid Aldubaikhy, *Member*, *IEEE*, Abdullah Alqasir, *Member*, *IEEE*, Xuemin (Sherman) Shen, *Fellow*, *IEEE*

***Abstract*—As an integral component of the space-air-ground integrated network (SAGIN), the low Earth orbit (LEO) satellite network has displayed immense potential in providing ubiquitous connectivity and broadband mobile communication. However, the intrinsic dynamics of LEO satellites pose unprecedented challenges in network management and service delivery. In this paper, we investigate the service function chain (SFC) orchestration in dynamic LEO satellite networks to achieve flexible and efficient service provision. Considering the service requirements and the limitations of network resources, we formulate the SFC orchestration problem as the integer nonlinear programming (INLP) problem for maximizing the service acceptance and the load fairness of satellites. Then, an efficient heuristic algorithm is proposed to solve this problem. Addressing the situation with frequent service requests, a graph attention network (GAT)-based approach with low complexity is also presented. Simulation results demonstrate that our proposed approaches outperform the benchmarks by a substantial margin in terms of load fairness and service acceptance. Besides, the proposed GAT-based approach shows its advantage in computation complexity, and exhibits robustness in unstable network scenarios with intermittent link interruptions.**

***Index Terms*—Low-Earth orbit (LEO) satellite network, load fairness, software-defined networking (SDN), service function chain (SFC) orchestration.**

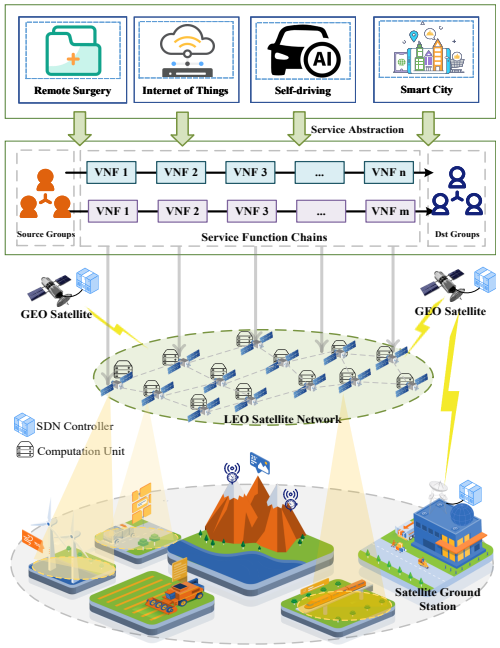

Fig. 1. An SDN/NFV-enabled LEO satellite network management framework.

I. INTRODUCTION

THE evolution of terrestrial networks has unlocked possibilities in service provision previously deemed unimaginable or even implausible. However, with the expanding of human activities, natural monitoring, demands on space security, etc., limitations are imposed on conventional terrestrial networks both practicably and financially. To fullfill the ubiquitous coverage, the space-air-ground integrated network (SAGIN) has attracted both industries and academia due to its advantage in network access anywhere and anytime for realizing 6G. The SAGIN is composed of space segments, air segments, and ground segments, where the low Earth orbit (LEO) satellite network plays a key role in the space segments and has experienced an unprecedented development [1], [2]. By the end of August 2023, SpaceX had launched more than 5,000 LEO satellites and had more than 2 million subscribers until September 2023. These LEO satellites, orbiting at altitudes typically ranging from 500 to 2000 kilometers, offer a unique blend of broad geographic reach and reduced latency in communication. Their relatively close proximity to the Earth's surface compared to geosynchronous equatorial orbit (GEO) satellites enables LEO networks to provide high-speed Internet with lower latency, making them ideal for real-time applications such as voice and video communications, online gaming, and other time-sensitive services. Furthermore, LEO satellite networks are particularly beneficial in bridging the digital divide by providing extraordinary network coverage in

This work was supported by the National Key Research and Development Program of China (2020YFB1807700). (*Corresponding author: Nan Cheng.*)

Jingchao He, Nan Cheng, and Zhisheng Yin are with the State Key Laboratory of ISN and School of Telecommunications Engineering, Xidian University, Xi'an 710071, China (e-mail: jchhe@stu.xidian.edu.cn; dr.nan.cheng@ieee.org; zsyin@xidian.edu.cn).

Haibo Zhou is with the School of Electronic Science and Engineering, Nanjing University, Nanjing 210023, China (e-mail: haibozhou@nju.edu.cn).

Conghao Zhou is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, N2L 3G1, Canada (e-mail: c89zhou@uwaterloo.ca).

K. Aldubaikhy and A. Alqasir are with the Department of Electrical Engineering, College of Engineering, Qassim University, Qassim, Saudi Arabia (e-mail: {khalid, a.alqasir}@qec.edu.sa).

Xuemin (Sherman) Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, N2L 3G1, Canada (e-mail: sshen@uwaterloo.ca).

remote and rural areas where terrestrial network infrastructure is limited or non-existent [3], [4]. This is crucial in facilitating emerging services like remote surgery, environment monitoring, disaster response communications, and self-driving, where traditional connectivity methods are challenging to implement.

As the number of LEO satellites continues to grow, the management and operation of the entire network become increasingly intricate and challenging [5]. To address this challenge, software-defined networking (SDN) has been proposed to separate the data layer and control layer, offering centralized network management. Concurrently, network function virtualization (NFV) technology proves its capability to abstract heterogeneous physical resources and provide a flexible approach in resource management [6]. Combining SDN and NFV technologies, the administration of LEO satellite networks achieves a level of programmability, flexibility, and scalability, and the service function chain (SFC) technology is also enabled. Our proposed network management framework is shown in Fig. 1, where SDN serves as the backbone to conduct system administering, service acceptance, and resource scheduling, and NFV is in charge of resource provisioning. As network functions like session border controllers, load balancers, or advanced functions like remote sensing, object detection, and mobile edge computing (MEC) arrive in networks, they are described into a specific sequenced virtual network functions (VNFs) chain firstly [7]–[9]. Then, these VNFs are embedded on satellites with virtual links chained together to serve as a service, which is defined as the SFC. Notably, the SFC is customizable to different user requirements, which reduces the hardware dependence and provides a potential solution for future on-demand network service provisioning.

Recently, several works on SFC have been presented on ground networks and SAGINs. In ground networks, researchers mainly concern the resource utilization in optical core networks and cloud-edge synergy scenarios [10]–[18]. Whereas for the SFC orchestration in SAGINs, researchers mainly focus on the coverage ability of networks and study the service provision in heterogeneous networks [19]–[24]. In [20], the SFC orchestration problem is presented to balance the resource utilization of both ground networks and non-terrestrial networks (NTNs). In [21], a large-scale heterogeneous network is considered and federated learning is utilized to cope with the data island problem in SFC orchestration. To date, only a few works have delved into SFC orchestration involving satellite networks [24]. Their focus primarily revolves around resource sharing and competition among various SFCs. Nevertheless, few works consider the network dynamics and load balance. In LEO satellite networks, the topology, propagation delay, and radio environment are time-varying, leading to situations where previously established orchestration strategies may no longer meet user requirements or could even disrupt services. Furthermore, existing studies often overlook the critical aspect of load balancing. When individual satellites handle an excessive number of services, it can lead to rapid saturation, reduced equipment and lifespan. Such issues have a direct and detrimental impact on the stability of the entire satellite constellation and severe network congestion, which leads to higher operational costs and lower service acceptance ratio and resource consumption [25]. Therefore, the SFC orchestration in LEO satellite networks must not only be efficient in service provision but also adaptable to the network's dynamic nature, taking into consideration the load on the global satellite network.

Deep reinforcement learning (DRL) has garnered significant attention as a promising technique for handling complex optimizations, particularly those that can be characterized as Markov decision processes (MDPs) [26], [27]. This enables rapid and astute decision-making in SFC orchestration, and considerable efforts have been dedicated [28]–[31]. However, the multilayer perceptron (MLP) in DRL operates on Euclidean space and faces challenges in spatial awareness and feature extraction of both nodes and edges in graphs, especially for time-varying LEO satellite networks. To overcome these drawbacks, graph neural network (GNN) has emerged as a powerful approach for representing complex relationships in graphs like social networks, transportation systems, and communication networks [16], [32]–[35]. Nonetheless, traditional GNNs like graph convolutional networks (GCNs) have limitations in handling varying importance of diverse resources for different SFCs. Additionally, existing GNN-based approaches on SFC orchestration are trained within a predetermined action space, heavily relying on the initial searching algorithm, which constrains the expression of feature extraction module and the exploration of reinforcement learning (RL) in return. Thus, there is an urgent need for advanced GNN architectures that can efficiently process complex and time-varying graph structure of LEO satellite networks, extract diverse node state information, and utilize diverse edge features in LEO satellite networks. Furthermore, exploring innovative RL approaches that break free from the limitations of pre-fetched action spaces will be crucial to enhancing the overall performance and scalability of SFC orchestration in dynamic LEO satellite networks.

Against the above challenges, in this paper, we investigate the SFC orchestration problem in LEO satellite networks. To maximize the service acceptance under limited network capacity and diverse service requirements, the SFC orchestration in LEO satellite networks is formulated as an integer nonlinear programming (INLP) problem. To solve this problem, we propose an efficient heuristic algorithm and a graph attention network (GAT)-based hierarchical RL approach with low complexity. Considering the dynamics and resource utilization in LEO satellite networks, we formulate the VNF migration problem to maintain the service continuity. Then, two solutions based on each SFC orchestration approach are presented to minimize the migration cost. Finally, extensive simulations are conducted to evaluate the performance of the proposed algorithms in terms of service acceptance, fairness of satellite load, robustness, etc. The main contributions are summarized as follows.

1) We propose an SDN/NFV-enabled network management framework to support multi-dimensional resource scheduling in LEO satellite networks. Based on this architecture, we formulate the SFC orchestration and migration into the INLP problems to maximize the service acceptance and fairness of satellite load in dynamic LEO

satellite networks.

2) Considering the satellite load and the service request, we present an efficient heuristic algorithm to search the path for the virtual link. Then, a greedy mechanism is utilized to embed each VNF. For the potential interruption of services caused by the movement of satellite constellation, a Tabu search (TS)-based algorithm is introduced to optimize the VNF migrations and ensure the continuity of service.
3) To cope with the situation with frequent service requests, a GAT-based hierarchical RL approach with low complexity is also proposed to schedule diverse SFCs in dynamic LEO satellite networks. Notably, the proposed GAT-based approach can achieve similar performance to that of the heuristic algorithm with lower computation complexity and is robust in unstable network scenarios.
4) Extensive simulation results are exhibited to evaluate the proposed algorithms in convergence, service acceptance ratio, and fairness. In addition, we perform our algorithms over several LEO satellite networks to evaluate their computation efficiency.

The remainder of this paper is organized as follows. Section II briefly reviews the related work. In Section III, we elaborate the system model in detail. The SFC orchestration problem and migration problem in dynamic LEO satellite networks are formulated in Section IV. Then, two approaches are proposed to solve the proposed problem in V and VI, respectively. Section VII evaluates the performance of the proposed algorithm through extensive simulations. Finally, Section VIII concludes this paper.

II. RELATED WORK

Recent works in SFC orchestration have been reviewed in this section. In the following, the works are broadly classified from the perspective of network scenarios.

In ground networks, some works mainly concern the resource utilization in optical core networks and cloud-edge synergy scenarios [10]–[18]. To cope with the increasing number and complexity of services in datacenters, it is proposed to reduce the service completion time and response time in service scheduling process, and a multi-swarm particle swarm-based optimization approach is presented to optimize the service provision [10]. Considering the potential failures in large-scale networks, [11], [12] study the SFC placement problem under availability and resource constraints. In [11], the disaster-resilient SFC provision is studied by path protection mechanism, and the power consumption and the spectrum usage are minimized. A sideway cross backup model is proposed in [12] to guarantee higher availability, and the VNF embedding is optimized to balance the resource consumption and reliability. Some studies propose the SFC parallelism to process partially-ordered services, then, the end-to-end service latency [13] or resource consumption [14] is minimized. Similar research is conducted in [15], where the delay and resource consumption are jointly minimized with the consideration of safety level of each VNFs. Researchers in [17], [18] focus on the SFC orchestration in MEC. In order to deliver dependable service provisioning in MEC, [17] presents to utilize the digital twin technology to maintain the status of VNFs in real-time and the SFC orchestration is optimized to minimize service costs. In [18], the SFC orchestration problem and routing scheduling in a hybrid cloud-edge synergy scenario are investigated to minimize the resource utilization and service latency.

Besides, many researchers introduce the SAGINs to provide network services in large-scale areas [19]–[24]. To maximize the long-term revenue-to-cost ratio and minimize the service delay in massive interconnection scenarios, a distributed resource management architecture is proposed and based on which, the access selection is optimized by using a distributed DRL algorithm [19]. Considering the real-time sparsely distributed service requests, the unmanned aerial vehicles (UAVs)-enabled reconfigurable network architecture is proposed in [22], where the trajectories of UAVs are optimized to minimize the average service delay. Based on multi-tier computing network in SAGINs, a multi-functional time expanded graph-based framework is proposed to characterize multiple computing functions for one mission flow [23], then, the computation resources, bandwidth, and storage resources are optimized to maximize the maximum flow. Above studies are mainly based on air-ground-integrated networks, and they optimize the resource scheduling to maximize the utilization of network nodes in relatively limited coverage. To further enhance the network coverage in remote areas and fulfill the user demands on ubiquitous connection, the role of LEO satellite networks is non-negligible.

However, only a few works investigate the SFC orchestration in satellite networks [24]. They mainly consider the resource sharing and competition among each SFC, and formulate the problem as a noncooperative game. Then, an adaptive play algorithm is utilized to find the Nash equilibrium. In LEO satellite networks, a concerning issue arises when individual satellites become overburdened with numerous services, leading to a rapid reduction in equipment lifespan. This directly affects the stability of the satellite constellation, degrades network performance, and ultimately incurs more operational costs. Additionally, the current studies on SFC provision primarily rely on search-based heuristic algorithms. These algorithms often employ a simplistic and repetitive searching mechanism, lacking a comprehensive understanding of the global network status. As a result, overall performance may suffer, leading to a decline in service quality and execution efficiency. Recently, DRL has demonstrated its efficacy in network resource scheduling [16], [30], [36]. However, the diverse and extremely dynamic structure of SAGINs causes MLP-based DRL to perform poorly or fail entirely. In contrast, directly modeling and capturing complex dependencies and interactions in graph-structured data by GNNs makes them effective for generalization. However, current GNN-based SFC orchestration methods are still ineffective since they can only use pre-fetched action spaces and limited feature extraction, which will incur a dramatic performance decline in LEO satellite networks.

In traditional network settings, SFC orchestration is often considered within a static or quasi-static network scenario, where network topology and channel status remain unchanged

over extended periods. In these settings, the focus is typically on minimizing or balancing resource consumption across heterogeneous network nodes, such as UAVs and BSs. However, in LEO satellite networks, the network topology and channel status are time-varying. Then, problems are raised when the initial orchestration strategy is broken or the requirements of users are violated by the movement of satellites. In previous studies, it was implied to reacquire user privacy data and re-establish customized VNFs in networks. While this approach may be feasible in scenarios with manageable end-to-end latency, it is problematic in dynamic LEO satellite networks. Here, the signaling interactions required for VNF re-establishment must traverse long distances, leading to unacceptable round-trip transmission delays and substantial communication resource wastage. Therefore, the issue of VNF migration becomes critical in LEO satellite networks, distinguishing it from traditional network settings. Furthermore, due to the relatively static nature of network topology in traditional SFC problems, many heuristic algorithms and MLP-based approaches are effective. However, the dynamics of LEO satellite network topology and channel status poses significant challenges. These conditions require enormous iterative processes for heuristic algorithms and challenge the effectiveness of MLP-based approaches in spatial awareness and feature extraction within graphs. Consequently, existing studies on SFC orchestration may be inefficient or even ineffective for dynamic LEO satellite networks. To provide service provision ubiquitously and continuously, a capable SFC orchestration approach considering both the service migration, satellite loads, and network dynamics is urgently desired.

III. SYSTEM MODEL

Fig. 1 illustrates an SDN/NFV-enabled LEO satellite network management framework, comprising three integral segments: the space network segment, the ground network segment, and the user segment. The space network segment forms the backbone of this architecture, featuring LEO satellite constellations for direct, low-latency communication and data relay. These constellations are augmented by GEO satellites, which provide a broader, more stable perspective for domain-level network supervision, essential for managing the dynamically changing network topology of LEO satellites. The ground network segment acts as a critical junction for service orchestration and overall network management. It is equipped with satellite ground stations that house satellite gateways and SDN controllers. These controllers utilize NFV technology to dynamically manage network resources and service flows, ensuring efficient and flexible network operation. The user segment, consisting of diverse user terminals, demands a range of services, from high-bandwidth applications like remote surgery to real-time requirements of self-driving vehicles. Each LEO satellite within the network is furnished with computation and communication units, enabling it to support a variety of network services. Collaboration between LEO and GEO satellites, facilitated through ground stations, overcomes the limited scope of individual satellites, providing a comprehensive global view and enhanced network management capabilities.

Consider an environmental monitoring use case within this framework, specifically for forest fire detection in areas lacking backhaul networks. While IoT devices such as thermal imaging cameras or multispectral scanners can be deployed, the voluminous data they generate in real-time often cannot be transmitted directly to environmental agencies. Upon receiving a service request from an environmental agency, the request is initially defined as a specific SFC tailored to meet their requirements. Potential VNFs in this scenario include a data compression function for minimizing less critical data, a load balancing function to streamline collected data and avert traffic congestion, and a deep packet inspection function to safeguard against data breaches or malicious interference. If the network's current state can accommodate this request, the SDN controller reserves a virtual link from the targeted area's LEO satellite to the one nearest the environmental agency. Concurrently, the pre-designed VNFs are sequentially embedded along this virtual link on the LEO satellite. Once the virtual link and VNFs are set up, the necessary data is transmitted until the monitoring task finishes. Subsequently, the resources utilized are released for other applications. In the following sections, we will introduce our network model and service model.

A. Network Model

In this study, we design the network model as a general Walker Star LEO satellite constellation, which comprises N homogeneous satellites evenly distributed in M LEOs. The physical network, denoted by $G = (\mathcal{F}, \mathcal{E})$, consists of the set of LEO satellites $\mathcal{F}$ and the set of physical links between satellites $\mathcal{E}$. Since network access selection is a significant and multifaceted research area encompassing various factors like channel status and elevation angle, we consider that users prioritize accessing the LEO satellite with the shortest line of sight distance and ignore the access process. Each LEO satellite is equipped with four transceivers, with two connected to intra-orbit satellites and the other two connected to the nearest inter-orbit satellites. The available bandwidth of the physical links between satellites n and m is denoted by $B_{n,m} \in \mathbf{B}$, and $\mathbf{B}$ is the set of available bandwidth of links. Furthermore, the channel delay $d_{n,m}$ is defined as the distance between the satellite pair (n, m) divided by the speed of light c, as expressed as

$$d_{n,m} = \frac{distance(n,m)}{c}, \forall (n,m) \in \mathcal{E}. \tag{1}$$

Without loss of generality, we assume the satellite orbit is circular, which means the eccentricity is 0 and the semi-major axis equals the radius of the orbit. The location of satellite n in Cartesian coordinates is denoted by $(p_{x,n}, p_{y,n}, p_{z,n})$, which is expressed as (2), where h is the orbit altitude of satellite n, R_e denotes the radius of the Earth, α denotes the inclination angle of satellite orbit, β denotes the right ascension of ascending node, τ denotes the true anomaly, and ξ denotes the argument of the perigee. Then, the distance between satellites n and m can be expressed as

$$\begin{aligned} distance(n,m) = [&(p_{x,n} - p_{x,m})^2 + (p_{y,n} - p_{y,m})^2 + \\ &(p_{z,n} - p_{z,m})^2]^{\frac{1}{2}}. \end{aligned} \tag{3}$$

$$\begin{pmatrix} p_{x,n} \\ p_{y,n} \\ p_{z,n} \end{pmatrix} = (h + R_e) \begin{pmatrix} \cos(\xi+\tau)\cos\beta - \sin(\xi+\tau)\cos\alpha\sin\beta \\ \cos(\xi+\tau)\sin\beta + \sin(\xi+\tau)\cos\alpha\cos\beta \\ \sin(\xi+\tau)\sin\alpha \end{pmatrix}. \tag{2}$$

As we focus on the communication and computation ability of satellites, the energy consumption in re-boosting is neglected, and the load of satellite n is expressed as

$$l_n = \zeta_1 c_n + \zeta_2 b_n, \tag{4}$$

where ζ_1, ζ_2, c_n, and b_n are the energy consumption weights of computation and communication, utilized computation resources, and utilized communication resources of satellite n, respectively. The set of load of LEO satellites is denoted by $\mathbf{l}$. Then, we utilize the Jain's fairness index to evaluate the fairness of load in LEO satellite networks, which is expressed as

$$f_t = \frac{\left(\sum_{n\in\mathcal{F}} l_n\right)^2}{|\mathcal{F}| \cdot \sum_{n\in\mathcal{F}} {l_n}^2}. \tag{5}$$

B. Service Modeling

As previously noted, the service model utilized in our system is the SFC, which is made up of a sequence of VNFs that are chained together in a predefined manner. To ensure successful service delivery, the sequential embedding of each VNF on the network node is necessary, and the end-to-end delay must fulfill the user's requirement. Before each decision-making interval, we assume that the arrived services have been collected and ready to be orchestrated, and the set them is denoted by $\mathcal{Q} = \{q | q = 1, 2, ..., |\mathcal{Q}|\}$. The bandwidth-demand service and the computation-demand service are the two sorts of services that are taken into account. The collection of source nodes and destination nodes is denoted by $\{s_q | q \in \mathcal{Q}\}$ and $\{d_q | q \in \mathcal{Q}\}$, respectively. For each service q, we define its VNF chain as $\mathrm{v}_q = \{v_i | i = 1, 2, \ldots, |\mathrm{v}_q|\}$, where v_i is the i-th VNF and $|\mathrm{v}_q|$ is the total number of VNFs. Denote the data flow between v_i and v_j by $\mathcal{E}_q = \{(v_i, v_j) | v_i, v_j \in \mathrm{v}_q, q \in \mathcal{Q}\}$.

Introduce the binary variable $x_{v_i,n,q} = 1$ to signify the embedding of VNF i of service q on node n, and $x_{v_i,n,q} = 0$ if not, and its solution vector is denoted by $\mathbf{x} = \{x_{v_i,n,q} \mid v_i \in \mathrm{v}_q, n \in \mathcal{F}, q \in \mathcal{Q}\}$. Similarly, the binary variable $y^{(n,m)}_{(v_i,v_j),q}$ takes the value 1 when virtual link (v_i, v_j) of service q is mapped onto physical link (n, m); otherwise, $y^{(n,m)}_{(v_i,v_j),q} = 0$. Correspondingly, the solution vector $\mathbf{y} = \{y^{(n,m)}_{(v_i,v_j),q} \mid (v_i, v_j) \in \mathcal{E}_q, (n, m) \in \mathcal{E}, q \in \mathcal{Q}\}$. We further denote the binary variable $z_q = 1$ to signify the reception of service q, and $z_q = 0$ otherwise. The solution vector $\mathbf{z}$ takes the form $\{z_q \mid q \in \mathcal{Q}\}$.

For users, the delay requirement on each SFC is compulsory to guarantee the quality of service (QoS) and quality of experience (QoE). In reality, the delay of SFC is mainly composed of communication delay and processing delay. Since the processing delay is related to the protocol, performance, and other uncontrollable factors, we consider the delay of each SFC is dominated by the communication [20], which is expressed as

$$t_q = \sum_{(n,m)\in\mathcal{E}} \sum_{(v_i,v_j)\in\mathcal{E}_q} y^{(n,m)}_{(v_i,v_j),q} d_{n,m}, \forall q \in \mathcal{Q}. \tag{6}$$

IV. PROBLEM FORMULATION

In this section, we investigate the SFC orchestration in dynamic satellite networks under the constraints of service provision, flow conservation, and network capacity. An INLP problem is formulated to maximize service acceptance and load fairness. Then, to maintain the SFC in dynamic LEO satellite topology, the VNF migration problem is formulated to minimize the migration costs.

A. Service Provision Constraints

This subsection presents the constraints of service provision. Before the service is delivered to the network, the source, destination, and VNF sequence are all predefined. Constraints C_1 and C_2 must be satisfied in order to guarantee that the initial and final VNFs are embedded on the source and destination, which is expressed as

$$C_1 : x_{v_1,s_q,q} = z_q, \quad \forall q \in \mathcal{Q}, \tag{7}$$

$$C_2 : x_{v_{|\mathrm{v}_q|},d_q,q} = z_q, \quad \forall q \in \mathcal{Q}. \tag{8}$$

As a received SFC, each of its VNFs is limited to embed on only one network node, which is expressed as

$$C_3 : \sum_{n\in\mathcal{F}} x_{v_i,n,q} = z_q, \quad \forall v_i \in \mathrm{v}_q, \quad \forall q \in \mathcal{Q}. \tag{9}$$

For each service, the transmission delay must be within the deadline T_q, which is expressed as

$$C_4 : t_q \le T_q, \quad \forall q \in \mathcal{Q}. \tag{10}$$

B. Flow Conservation Constraints

Since the SFC is sequenced VNFs chained by data flow, the flow conservation is critical to ensure the data flow is properly processed. Consequently, the flows in should equal to the flows out, which is expressed as

$$C_5 : \sum_{m\in\mathcal{F}} y^{(n,m)}_{(v_i,v_j),q} - \sum_{m\in\mathcal{F}} y^{(m,n)}_{(v_i,v_j),q} = x_{v_i,n,q} - x_{v_j,n,q}, \quad \forall (v_i, v_j) \in \mathcal{E}_q, \forall n \in \mathcal{F}, \forall q \in \mathcal{Q}. \tag{11}$$

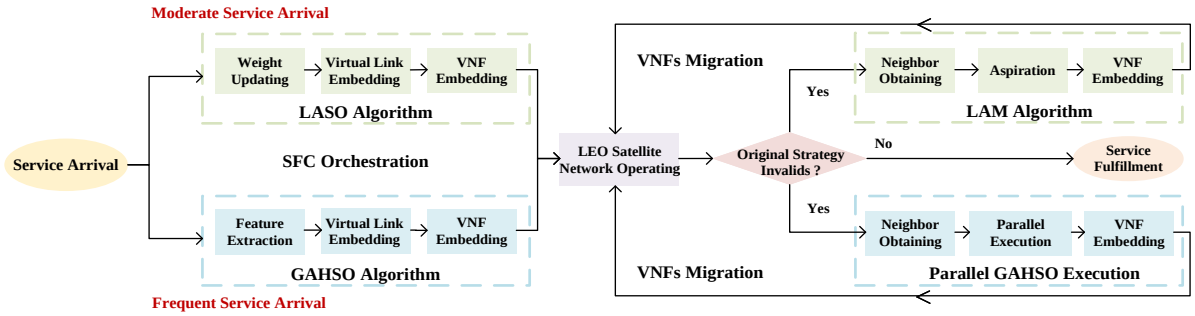

Fig. 2. The workflow of algorithm execution.

C. Network Node Constraints

For each satellite, its computation and communication resources are limited, and the allocated network resources cannot exceed the satellites' capacity, which is expressed as

$$C_6: \sum_{q\in\mathcal{Q}}\sum_{v_i\in\mathbf{v}_q} x_{v_i,n,q}c_{v_i,q} \le C_n, \quad \forall n\in\mathcal{F}, \tag{12}$$

$$C_7: \sum_{q\in\mathcal{Q}}\sum_{(v_i,v_j)\in\mathcal{E}_q} y^{(n,m)}_{(v_i,v_j),q} b_q \le B_{n,m}, \quad \forall (n,m)\in\mathcal{E}, \tag{13}$$

where $c_{v_i,q}$ denotes the computation utilization of VNF i of service q, b_q denotes the communication utilization of service q, $C_n \in \mathbf{C}$ denotes the computation capacity of satellite n, and $\mathbf{C}$ is the set of computation capacity of LEO satellites. C_6 constrains the computation resource utilization, and C_7 constrains the bandwidth utilization.

D. SFC Orchestration and Migration Problem

The SFC orchestration is optimized to maximize the service acceptance and fairness of satellite load. An INLP problem is formulated with the consideration of constraints $C_1 - C_7$, which is expressed as

$$\begin{aligned} P1: \max_{\mathbf{x},\mathbf{y},\mathbf{z}} & \sum_{q\in\mathcal{Q}} z_q + \gamma_1 \frac{\left(\sum_{n\in\mathcal{F}} l_n\right)^2}{|\mathcal{F}|\cdot\sum_{n\in\mathcal{F}} {l_n}^2} \\ s.t. & C_1 - C_7, \\ & C_8: \mathbf{x},\ \mathbf{y},\ \mathbf{z} \in \{0,1\}, \end{aligned} \tag{14}$$

where γ_1 is the weight of fairness to balance the service acceptance and fairness of satellite load.

In LEO satellite networks, the satellites move very fast, which changes the network topology and channel status frequently, rendering previously optimal strategy suboptimal or even broken over time. To maintain the SFC continuity and fulfill users' QoS requirements in such a dynamic network scenario, live VNF migration is unavoidable. However, the customized VNFs combine user privacy and dedicated data, and end-to-end retransmission will bring intolerable delay and communication resource waste, which can be considered as the migration costs. To maintain the continuity of services, maximize the load fairness, and minimize the migration costs, we formulate the VNF migration problem when the service is interrupted by the movement of satellite networks, which is expressed as

$$\begin{aligned} P2: \min_{\mathbf{x},\mathbf{y},\mathbf{z}} & \gamma_2 \sum_{q\in\mathcal{Q}}\sum_{v_i\in\mathbf{v}_q} h_{v_i,q} - \sum_{q\in\mathcal{Q}} z_q - \gamma_1 \frac{\left(\sum_{n\in\mathcal{F}} l_n\right)^2}{|\mathcal{F}|\cdot\sum_{n\in\mathcal{F}} {l_n}^2} \\ s.t. & C_1 - C_7, \\ & C_8: \mathbf{x},\ \mathbf{y},\ \mathbf{z} \in \{0,1\}, \end{aligned} \tag{15}$$

where γ_2 is the cost weight of VNF migration, $h_{v_i,q}$ is the migration decision of VNF i of service q, where $h_{v_i,q} = 1$ denotes the VNF i will migrate to a new network node, $h_{v_i,q} = 0$ otherwise. The weights of VNF migration and fairness ensure the continuity of each SFC as much as possible. In P2, we minimize the migration of VNFs and maximize the load fairness of satellite networks on the premise of ensuring service continuity. For each service, it will occupy a specific block of resources for a period of time. Thus, z_q implies the service is supported continually and $z_q = 0$ when P2 is triggered signifies the failure in migration process.

V. LASO-LAM Approach

As previously mentioned, our study focuses on the dynamic provisioning scenario of SFC orchestration, where requests arrive and expire spontaneously, and are served sequentially based on their arrival time. In this paper, we consider two distinct scenarios, as shown in Fig. 2, based on the service arrival frequency: one with a normal frequency of service arrivals and the other with a higher frequency of service arrivals, which are termed as moderate service arrival and frequent service arrival, respectively. In cases of moderate service arrival, a load-aware SFC orchestration algorithm, i.e., LASO algorithm, is employed to make efficient decision in manageable scale of service requests. Conversely, during periods of frequent service arrival, which signal a heightened demand on the network, a GAT-based hierarchical RL approach with low inference complexity, i.e., GAHSO algorithm, is utilized. This algorithm is more adept at managing scenarios with massive user request and facilitating rapid decision-making. Subsequently, the system checks if the original strategy remains valid with the LEO satellite networks operating. If the strategy remains effective until the completion of service provision, the service is deemed successfully executed; otherwise, the SFC migration approach is executed to recalibrates the SFC

orchestration strategy. In situations of moderate service arrival, a TS-based migration algorithm, called LAM, is employed for VNF migration. Since GAHSO Algorithm is able to be executed parallelly in GPU, the action space in migration can be exhaustively and efficiently searched. The details of the algorithms are presented separately in Section V and Section VI.

In this section, we present the LASO algorithm to derive near-optimal solutions for problem P1. Moreover, LAM is proposed to sustain service continuity and minimize VNF migration in dynamic satellite networks. The details are as follows.

The SFC orchestration can be decoupled as VNF embedding and virtual link embedding [20]. We find the optimal virtual routing path with the consideration of fairness of satellite load, as shown in Algorithm 1. The available link capacity and the end-to-end delay should meet the service requirements, which is denoted by $Req(q) = \{s_q, d_q, T_q, \mathrm{v}_q\}$. To guarantee the adequate channel capacity, we introduce the indicator function into the link weight for each service, denoted by $\mathbb{I}\left(b_{n,m} - b_q\right)$, and $b_{n,m}$ is the available bandwidth in link (n, m). We design the weight of physical link (n, m) as

$$W_q(n,m) = \frac{d_{n,m}\mathbb{I}\left(b_{n,m} - b_q\right)}{\exp\left(\theta_0\left(l_n + l_m\right)/2\right)}, \forall q \in \mathcal{Q}, \forall (n,m) \in \mathcal{E}, \tag{16}$$

where θ_0 is the load fairness factor and negatively correlated to the importance of the value of load. The load fairness factor balances the service delay and satellite load. When $\theta_0 = 0$, Algorithm 1 will generate the path with the lowest end-to-end delay, and when $\theta_0 = 1$, the load of connected satellites is valued. Initially, the value of θ_0 is set to 1 to search the routing path with maximal load fairness. Based on the updated weight by (16), the potential routing path is generated by the Dijkstra algorithm, and Algorithm 2 with a greedy mechanism is enabled to schedule the VNF embedding. Potential nodes with maximum computational capability are picked to deploy each VNF of q. When the prospective nodes of the produced path from Algorithm 1 are unable to instantiate all VNFs, the embedding result is marked as a fault. Then, set $\theta_0 = \theta_0 - \Delta_\theta$ and continue the searching loop until the halting criteria is met. If $\theta_0 < 0$ in execution, it is implied that the orchestration process has failed and the service has to be blocked.

Algorithm 1: LASO algorithm

Input: Service requirements $Req(q)$, available computation resources $\mathbf{C}$, available communication resources $\mathbf{B}$, and current network graph G
Output: SFC orchestration strategy

```
Set θ0 ← 1;
while θ0 ≥ 0 do
    Calculate the weight of edges in G by (16);
    Utilize the Dijkstra algorithm and obtain the shortest path;
    Embed the VNFs of service q by Algorithm 2;
    if t_q ≤ T_q and embeddingIndex ≠ fault then
        Output the routing path and embedding strategy u_q;
        Break the loop;
    else
        θ0 = θ0 − Δ_θ;
if θ0 < 0 then
    Block the service q;
```

Algorithm 2: VNF Embedding algorithm

Input: Service requirements $Req(q)$, potential routing path, and available computation resources $\mathbf{C}$
Output: Embedding strategy u_q,

```
Choose the candidate network node CandidatesList with maximal computation resources ;
for VNF i ∈ v_q do
    if Available resource is enough then
        Embed the VNF i in CandidateList(i);
        update the VNF embedding strategy u_q;
    else
        Set embeddingIndex ← fault ;
        Break the loop;
if All VNFs are embedded successfully then
    Set embeddingIndex ← true;
else
    Set embeddingIndex ← false;
```

In order to maintain continuous support for SFCs within the dynamic network environment, Algorithm 3 springs into action when the original strategy becomes invalid. In this algorithm, a TS-based migration mechanism is utilized to minimize the VNF migration costs and maximize the load fairness when the initial embedded virtual link of q is broken or its delay violates the requirements. The collection of migration decision of each VNF in service q at k-th iteration is denoted by $g_{k,q} = \{h_{v_i,q} | v_i \in \mathrm{v_q}, q \in Q\}$, whose length equals $|\mathrm{v}_q|$. Each of its elements is initialized to zero, symbolizing that the initial migration decision for all VNFs stands as False. Subsequently, we obtain the neighbors of $g_{k,q}$, and for each decision g' in the set of neighbors, the corresponding network nodes are dropped within a replicated version of the initial network. Then, we compute the edge weight by (16) and splice the link of remaining VNFs by the Dijkstra algorithm. In parallel, the migrated VNFs are seamlessly integrated by employing Algorithm 2, and the migration cost is obtained. There, the Tabu list is defined as a caching list to store the update vector in a period of time to avoid similar strategies being generated. Additionally, the influence is utilized as our aspiration criteria to avoid overload nodes are chose frequently. As the procedure continues, the cost value of each migration decision is calculated. The potential solution, denoted by u_q^*, takes shape through continuous updates when its cost is below the minimum and update vector is not in the Tabu list. Finally, the procedure stops if the iteration exceeds the maximum value K or u_q^* remains unchanged in several iterations successively, and the migration strategy is obtained if u_q^* is not null.

Algorithm 3: LAM algorithm

Input: Service requirements $Req(q)$, initial embedding strategy u_q, available computation resources $\mathbf{C}$, available communication resources $\mathbf{B}$, and current network graph G

Output: SFC migration strategy

Set $k \leftarrow 0$;
Set $g_{k,q} \leftarrow zeros(|\mathrm{v_q}|)$;
while $k < K$ **do**
  Set $G' \leftarrow G$;
  $Neighbors(g_{k,q}) \leftarrow findNeighbors(g_{k,q})$;
  for g' *in* $Neighbors(g_{k,q})$ **do**
    for *Network node* n *in* g' **do**
      Drop n in G;
    Calculate the weight of G' by (16);
    Splice the link of remaining VNFs by the Dijkstra algorithm in G';
    Obtain the VNF embedding strategy u' by Algorithm 2;
  Obtain the migration strategy in $Neighbors(g_{k,q})$ with minimal migration cost;
  if $migrationCost(u'_q) < migrationCost(u^*_q)$ *and* u'_q *not in* $TabuList$ **then**
    Set $u^*_q \leftarrow u'_q$;
    update $TabuList$;
if u^*_q *is not null* **then**
  Set $migrationIndex \leftarrow true$;
  Set $u_q \leftarrow u^*_q$;
else
  Set $migrationIndex \leftarrow false$;

For Algorithm 1, the computation complexity mainly comes from weight computation and the Dijkstra algorithm. In this part, we analyze the worst case to evaluate the performance of LASO algorithm. Firstly, the worst-case time computation complexity of $while$ loop is $T(\lceil\theta_0/\Delta_\theta\rceil)$. Then, the computation complexity in each iteration is $O(|\mathcal{E}| + N^2)$. Similarly, the complexity of Algorithm 2 is $O(\log(|\mathrm{v}_q|)N)$. Thus, the total complexity of our orchestration algorithm is $O(\lceil\theta_0/\Delta_\theta\rceil[N^2 + |\mathcal{E}| + \log(|\mathrm{v}_q|)N])$.

VI. GAHSO Algorithm

Currently, forecasting service requirements and user volumes accurately remains an elusive challenge in both industry and academia [37]. This necessitates consideration for scenarios characterized by an abundance of users or a high frequency of service requests. However, the proposed algorithm in Section V based on the Dijkstra algorithm will experience a quadratically increase in computation as the number of LEO satellites increases. Thus, it may result in decreased response times and heightened resource consumption in controllers, potentially hindering real-time or time-sensitive applications in future large-scale LEO satellite networks. To this end, we propose a RL-based algorithm with low complexity, tailored to effectively manage situations involving high-intensity business volumes. In the following, the dynamic SFC orchestration is first modeled as an MDP. Then, a GAT-based extraction module is utilized to extract the multi-dimensional attributes of both the LEO satellite networks and service requests. Finally, we propose a GAT-based A2C algorithm to solve the proposed problem. Specifically, a hierarchical structure is utilized to shape the sparse reward and judge the actions, and the proposed approach is named as GAT-based hierarchical A2C SFC orchestration algorithm, i.e., GAHSO algorithm.

A. Modeling Dynamic SFC Orchestration

To effectively capture the dynamic transitions between network states during service provisioning, we model the transition of SFC orchestration by MDP. Specifically, it is defined as a tuple $<S, A, R, P>$, where S denotes the state space in the SFC orchestration process, which is composed of information of the LEO satellite networks and the current orchestration process. A denotes the available action space, and R denotes the reward for each action and state. The reward of SFC orchestration evaluates and provides quantitative feedback on the agent's actions in a given state, indicating whether they align with the task objectives. P plays a crucial role in RL by describing the probability of the agent transitioning from one state to another when taking a specific action. It models the dynamic changes in the environment.

The state in this MDP at time t, i.e., $s_t \in S$, is comprised of the information of the current orchestrated service, and the LEO satellite network containing the topology and remaining resources of G. For the information of currently orchestrated service, it is attached to each network node to mark the source, the destination, and the path of virtual link passed by. The action $a_t \in A$ at time t is the decision of agent determined by s_t and the policy function, it influences the network state of next time, i.e., s_{t+1}. After executing a_t, an instant reward $r_t \in R$ is generated by the reward function. The state transition of determination process of this MDP can be expressed as $<s_t, a_t, r_t, s_{t+1}>$ until the interruption is activated if the orchestration is finished or the number of algorithm executions exceeds the threshold. To be specific, the agent starts from the source of the current service, and each action is the routing decision of the virtual link. Only when the agent moves to the destination, i.e., the virtual link is embedded, and every VNF is embedded along that link, the orchestration is successful. In previous studies, the SFCs are usually orchestrated based on the available routing paths obtained by the Dijkstra algorithm, which neglects the overall network status and is restricted to the available action sets. In our proposed algorithm, we break free from this limitation and expand the action spaces to all available neighbor nodes.

In the following, we will introduce the basic element of RL in our proposed problem in detail.

1) State Representation: The state in this model should represent all the useful information relevant to service orchestration, and it is compromised of the network feature and service feature. The network feature contains the network topology, remaining computation resources, remaining

communication resources, and the load of each satellite of G, which is part of the input of graph embedding module. The service feature encapsulates service information, i.e., the service type, the source, the destination, the traversed path, and the position information of agent, which is denoted by $I_t(q) = Req(q) \cup \{[a_0, ..., a_{t-1}]\}$. Then, the state at t is denoted by $s_t = \{G_t, \mathbf{C}_t, \mathbf{B}_t, \mathbf{l}_t, I_t(q)\}$, where the subscript is utilized to signify the time in orchestration process.

2) Action Definition: During the orchestration process, the action impacts the potential VNF embedding and virtual link routing significantly, thus exerting a substantial influence on the overall orchestration of the SFC. In this part, we adopt a flexible but simple strategy compared with existing studies by allowing the agent to explore from all accessible neighbors, i.e., the action space contains all neighbors of the agent's current position. This shift away from predefined action sets provides greater adaptability and resilience, requiring only a mask layer to hide the non-neighbor nodes. Specifically, the action is denoted by an one-hot vector, i.e., $a_t = \{a_t^0, a_t^1, ..., a_t^n, ..., a_t^{N-1}\}$, where $a_t^n = 1$ denotes n is selected as the next hop of virtual link; otherwise, $a_t^n = 0$.

3) Reward Description: When devising the reward function for RL in SFC orchestration, it is essential to consider various factors such as rewarding successful orchestration, optimizing performance metrics, resource conservation, and avoiding invalid actions. Ensuring adaptability, stability, and compliance are key objectives, achieved by balancing objectives and adhering to principles of sparse rewards and long-term goals. As SFC orchestration is known to have sparse rewards, shaping the reward function is imperative. Accordingly, we design distinct reward functions for the determination process and the final state. During the determination process, the action's compliance with resource capacity constraints and service provisioning is vital. The instant reward before the final state is defined as

$$r_t = \begin{cases} -\theta_1 d_t - \theta_2 h_{a_t} - \theta_3 l_{a_t}, & \text{if action } a_t \text{ is valid,} \\ -\alpha_1, & \text{if action } a_t \text{ is invalid,} \end{cases} \tag{17}$$

where h_{a_t} denotes the hops of the shortest path to the destination, and θ_1, θ_2, and θ_3 serve as coefficients for delay from the previous network node to the next node, the number of shortest hops to the destination of service q, and the load of next node, respectively. $-\alpha_1$ represents the penalty for constraint violation, and it is set as a large constant value to deter the agent from making invalid determinations. When the action is valid, the distance to the destination, the delay, and the load of the satellite are assessed meticulously.

Similarly, the reward function in the final state should evaluate the overall SFC orchestration process, encompassing constraint violations, service provisioning, and network fairness. Compared with the optimization objective in Section IV, the reward function in this section is expected to provide a more precise, logical, and meticulous assessment of the final decision. Thus, we devise the reward function for the final state in three cases: successful orchestration, successful orchestration with a violated latency, and failed orchestration (i.e., the agent fails to arrive at the destination), which is expressed as

$$r_t = \begin{cases} \beta_0 + \gamma_1 f_t, & t_q \le t_r, \\ \beta_1 - \theta_4 t_q + \gamma_1 f_t, & t_q > t_r, \\ -\alpha_2 - \theta_5 h_{a_t} - \theta_6 n_{embed} + \gamma_1 f_t & \text{orchestration fails}, \end{cases} \tag{18}$$

where β_0 and β_1 represent the reward values of SFC orchestration, with β_0 being greater than β_1 to promote current orchestration and penalize routing paths violating the delay constraint. f_t denotes the current value of fairness calculated by (5). θ_4 is introduced to promote the delay reduction when the delay constraint is violated. α_2 serves as the penalty value for failed orchestration when the determination process exceeds the maximum action size or some VNFs fail to be embedded. n_{embed} denotes the number of VNFs not embedded successfully, while θ_5 and θ_6 represent coefficients for penalties due to constraint violation.

B. GAT for Feature Extraction in SFC orchestration

Unlike MLPs that are apt for processing vector-based data, GNNs offer the ability to effectively handle graph-structured data, such as social networks, knowledge graphs, and molecular structures. Furthermore, GNNs exhibit context awareness, transfer learning capabilities, and strong generalization performance. This enables knowledge transfer and reuse by training on one graph and transferring the learned knowledge to other graphs with similar structures, which is beneficial in adapting dynamic LEO satellite networks and resisting emergencies in unstable network scenarios. Based on the GNN, GAT introduces the concept of attention mechanisms, allowing for non-uniform node interactions and adaptive feature aggregation. By dynamically learning attention weights, GAT can capture the varying importance of nodes and effectively aggregate features from neighboring nodes. This flexibility and adaptability enable GAT to focus on important information and outperform traditional GNNs.

In this paper, we utilize the GAT as our feature extraction module to extract the information of LEO satellite networks and strengthen the expression of service requirements in message passing. The structure of our proposed network is illustrated in Fig. 3. The feature of each network node is denoted by $\mathbf{f}_i$ $(i \in \mathcal{F})$, which is a $1 \times F_1$ vector containing remaining computation resources and the indicator index of routing, and the matrix of node features is $N \times F_1$. Similarly, we denote the feature of edges by a $|\mathcal{E}| \times F_2$ matrix, and F_2 is the dimension of edge features including the delay and available transmission resources. The proposed graph has a multi-layer neural network, and k is superscript to denote the layer of network output. The detail of feature extraction is shown as follows.

For features of each node $i \in \mathcal{F}$, they are transformed into a new representation space through a linear transformation layer. For each node, the features are mapped to the input of a linear transformation matrix denoted by Θ_x, which is expressed as

$$\mathbf{f}_i^{(1)} = \mathbf{\Theta}_x \mathbf{f}_i^{(0)}, \tag{19}$$

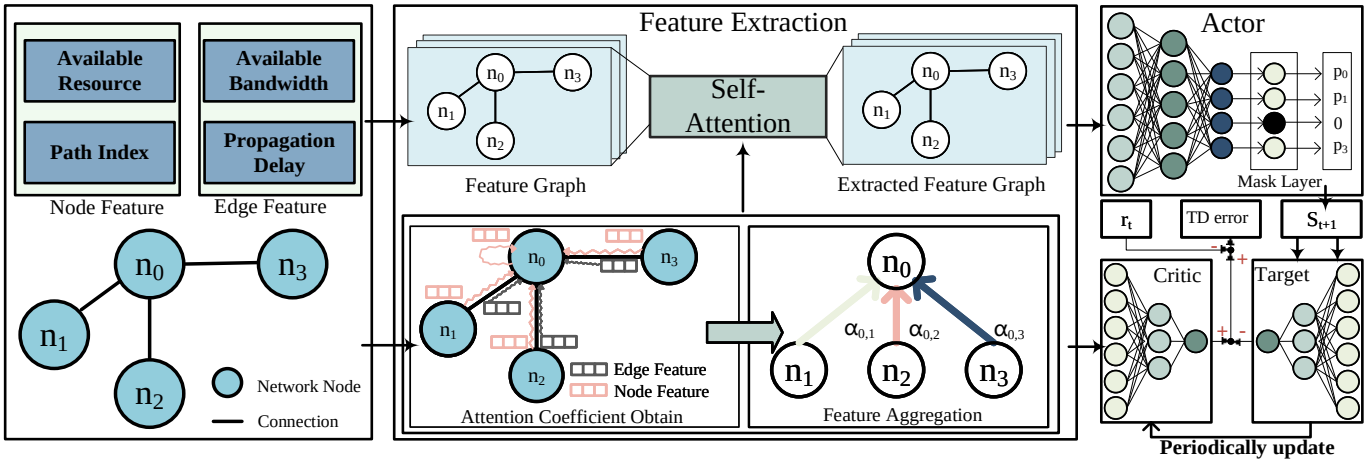

Fig. 3. The structure of GAHSO algorithm.

where $\mathbf{f}_i^{(1)}$ is the output of linear transformation, and in the following, we define the output of k-th layer network by a superscript. Similarly, the edge features can be transformed using another linear transformation matrix $\mathbf{\Theta}_e$. For each edge $(i,j) \in \mathcal{E}$, the transformed edge feature is computed as

$$\mathbf{e}_{i,j}^{(1)} = \mathbf{\Theta}_e \mathbf{e}_{i,j}^{(0)}, \tag{20}$$

where $\mathbf{e}_{i,j}^{(1)}$ is the output of edge feature of linear transformation.

Next, attention coefficients $\alpha_{i,j}$ are computed based on the transformed node features and edge features. Additionally, an activation function is utilized, i.e., $\sigma = \mathrm{LeakyReLU}()$. Then, the coefficients are normalized by a sigmoid function. For each node i and its neighboring nodes j, the attention coefficient $\alpha_{i,j}$ in the k-th layer is computed as:

$$\alpha_{i,j}^{(k)} = \frac{\exp(\sigma(\mathbf{a}^T[\mathbf{f}_i^{(k)} \| \mathbf{f}_j^{(k)} \| \mathbf{e}_{i,j}^{(k)}]))}{\sum_{n \in \mathcal{N}(i)} \exp(\sigma(\mathbf{a}^T[\mathbf{f}_i^{(k)} \| \mathbf{f}_n^{(k)} \| \mathbf{e}_{i,n}^{(k)}]))}, \tag{21}$$

where $\mathbf{a}$ is a learned attention weight vector, $\|$ represents the concatenation operation, and $\mathcal{N}(i)$ denotes the set of neighborhood of node i.

Finally, the node features are aggregated based on the computed attention coefficients. For each node i, the aggregated feature representation is computed as

$$\mathbf{f}_i^{(k)} = \alpha_{i,i}^{(k-1)} \mathbf{\Theta} \mathbf{f}_i^{(k-1)} + \sum_{j \in \mathcal{N}(i)} \alpha_{i,j}^{(k-1)} \mathbf{\Theta} \mathbf{f}_j^{(k-1)}, \tag{22}$$

where the aggregation step combines the neighboring node features and their corresponding edge features with respective attention coefficients.

In the output layer of graph embedding, an $N \times F_{hidden}$ matrix is generated, where F_{hidden} denotes the dimension of the hidden feature. So far, the features of the network and service are extracted and they can be pass forward to the next layer or flattened to a one-dimensional vector for further operation.

C. Hierarchical A2C-Based RL for Decision-Making

Compared with traditional problems like travelling salesman problem (TSP) or LunarLander-v2 in gym, the SFC orchestration is a multi-step and sparse-reward mission, and its action is difficult to be measured until the agent arrives at the destination of the current service. Moreover, the increasing number of network nodes in LEO satellite network improves the network capacity and incurs intractable action space conversely, which impedes the convergence of RL. As a result, we utilize a hierarchical actor-critic (AC)-based architecture for decision-making in SFC orchestration.

In AC algorithms, the critic network plays a crucial role in offering valuable feedback on the quality of actions chosen by the policy network. This feedback serves as a basis for the policy network to improve and optimize its actions effectively. A proficient critic network is capable of accurately reflecting the potential future outcomes, thereby facilitating precise corrections to the policy network's decision-making process. However, value estimates derived from critic network in SFC orchestration often exhibit high variance due to limited samples and discrete reward functions. This heightened variance may lead to unstable parameter updates and sluggish convergence. To address this challenge, the target network is introduced to serve as a stable criterion for policy network and is periodically updated with the weights from the value function network to attain enhanced accuracy.

The procedure is described in Algorithm 4. When a new SFC request q arrives, the network orchestrator records the state of current LEO network and the requirements of q. The information of service q containing the source, destination, service type, passing path (or potential passing path) together with the remaining computation resources are concatenated as the node features. The edge features are composed of normalized remaining bandwidth and propagation delay. Integrated with the node features and edge features, the input of GAT feature extraction layer at t is obtained, which is denoted by G_t. Additionally, a one-hot vector is introduced to indicate the current position in determination. Integrated with both G_t and the position vector, the information of s_t is obtained. The actor network observes the state s_t and outputs

an N-dimension vector, where each element in this vector is conducted as the score of each LEO satellite. Mask the non-neighbor nodes with negative infinity and input the result into an activation function, the determination for the next hop in s_t is generated. Subsequently, the reward r_t and new state s_{t+1} is obtained after executing a_t. To reduce the error incurred by bootstrapping, input s_t and s_{t+1} into critic network and target network, separately. Then, the critic network generates the state value $V(s_t|\theta_c)$, and the target network generates the state value of s_{t+1}, i.e., $V(s_{t+1}|\theta_t)$, where θ_c and θ_t are the parameters of critic network and target network, parallelly. The temporal difference error (TD-error) A_t is expressed as

$$A_t = V(s_t|\theta_c) - (r_t + \mu V(s_{t+1}|\theta_t)), \tag{23}$$

where μ is the discounted factor for long-term return. Then, the loss function and update gradient of critic network at t are derived as

$$L(t, \theta_c) := \frac{1}{2} A_t^2, \tag{24}$$

$$\nabla L(t, \theta_c) = \gamma_c A_t \cdot \nabla_{\theta_c} V(s_t|\theta_c), \tag{25}$$

where γ_c is the learning rate of critic network. Similarly, the update gradient of actor network [38] is derived as

$$\nabla L(t, \theta_a) = \gamma_a A_t \cdot \nabla_{\theta_a} \ln \pi(a_t|s_t; \theta_a), \tag{26}$$

where $\pi_(a_t|s_t; \theta_a)$ denotes the policy function of actor network, and γ_a is the learning rate of actor network. Until now, the gradient of both actor network and critic network is obtained, and they are accumulated until the orchestration of current service request is finished. Then, Algorithm 2 is utilized to embed the VNFs. In the determination process, the parameters are not updated, because the orchestration of SFC is not dependent on a single step but the whole determination process, and it will fail if the total delay is violated or it does not arrive at the destination. Therefore, we utilize a hierarchical update strategy to update the parameters once an orchestration process. Additionally, the proposed GAHSO approach is able to be processed parallelly in GPU, and all possibilities in VNF migration can be examined simultaneously. Hence, the strategy with minimal migration cost will be selected as the final decision for P2, directly.

Algorithm 4: GAHSO Training

Input: Service requirements $Req(q)$, initial available computation resources $\mathbf{C}$, initial available communication resources $\mathbf{B}$, and initial network graph G

Output: Trained policy network $\pi(\theta_a)$ and value function network $V(\theta_c)$

Initialize the parameters of policy network $\pi(\theta_a)$ and value function network $V(\theta_c)$ with random weights;
Synchronize the parameters of target network $V(\theta_t)$ with $V(\theta_c)$;
Initialize exploration parameters;
for *episode = 1 to* $EPISODE$ **do**
  Reset the network to the initial state G;
  Reset gradient of actor network and critic network as ∇L_a and ∇L_c;
  Set $flag \leftarrow 0$;
  for *each newly-arrived service request* q **do**
    Integrate the information of q and G;
    Set $t \leftarrow 0$;
    Get initial state s_t;
    while *service is not orchestrated successfully and* $t \leq K$ **do**
      Sample action a_t from the policy network π_t;
      Execute action a_t, and observe reward r_t and next state s_{t+1};
      Calculate the update gradient of critic network and actor network by (25) and (26), separately;
      $\nabla L_c \leftarrow \nabla L_c - \nabla L(t, \theta_c)$;
      $\nabla L_a \leftarrow \nabla L_a + \nabla L(t, \theta_a)$;
      $t = t + 1$;
    Update policy network π using gradient ascent: $\theta_a \leftarrow \theta_a + \gamma_a \nabla L_a$;
    Update value function network V using gradient descent: $\theta_c \leftarrow \theta_c + \gamma_c \nabla L_c$;
    if $flag = FLAG$ **then**
      Update target networks;
      Set $flag \leftarrow 0$;

D. Time Complexity Analysis

The time complexity of the GAHSO is analyzed as follows. In the proposed approach, the computation complexity is mainly incurred by inference in actor network, i.e., the GAT network and MLP. As mentioned before, N denotes the number of network nodes in LEO satellite networks, $|\mathcal{E}|$ denotes the number of edges in networks, F_1 denotes the feature of each node, F_2 denotes the dimension of edge features, and F_{hidden} denotes dimension of hidden features. The computation complexity in feature extraction contains the linear transformation, attention coefficient computation, and feature aggregation. In linear transformation, the required number of floating-point operations is $O(F_1 F_{hidden} N)$ and $O(F_2 F_{hidden}|\mathcal{E}|)$ for network nodes and edges, separately. In attention coefficient computation, (21) maps a $3 \times F_{hidden}$ matrix to a real number value if the attention head is one. Then, the required number of floating-point operations in this process is $O(3F_{hidden}|\mathcal{E}|)$. In feature aggregation, the operation type is mainly matrix-vector weighted summation, which is not as complex as multiplication and can be neglected. Besides, the activation operation also has a lower complexity and can be ignored [39]. Generally, the MLP in actor network contains a three-layer neural network, and the dimension of the middle layer is denoted by F_{m1} and F_{m2}. Similarly result for MLP network can be derived and whose worst-case time complexity is $T(F_{hidden}F_{m1}N + F_{m1}F_{m2} + F_{m2})$. Consequently, the complexity is $O(F_1 F_{hidden} N + F_{hidden} F_{m1} N + \log(|\mathrm{v}_q|)N + F_2 F_{hidden}|\mathcal{E}| + 3F_{hidden}|\mathcal{E}|)$. With the escalating scale of LEO satellite networks, it becomes evident that GAHSO presents a more efficient computation complexity when juxtaposed with

LASO, which is also verified in our simulations.

TABLE I
SIMULATION SETTINGS

Parameter	Value
The number of satellites in the network N	25-200
The height of satellite orbit R_e	2000 km
CPU capacity of each satellite C_n	400-600 CPU unit
The required bandwidth of each service b_q	20-80 MHz
The required CPU units of each VNF $c_{v_i,q}$	50-200 CPU unit
Bandwidth between each satellite $B_{n,m}$	200 MHz
Maximum of action size K	20-30
Learning rate of actor network γ_a	0.00001
Learning rate of critic network γ_c	0.00001
The discounted factor for long-term return μ	0.9999
The dimension of network node features F_1	128
The dimension of edge features F_2	128
The feature of hidden features F_{hidden}	128
The speed of light c	$2.99 \times 10^8 m/s$

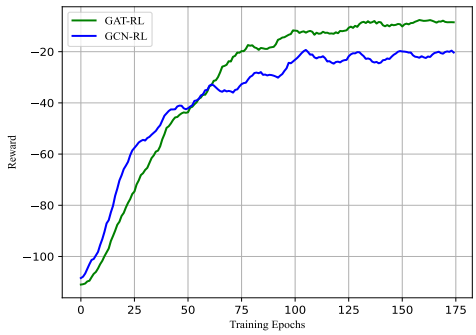

Fig. 4. Convergence trend of the reward with the training epoch.

VII. PERFORMANCE EVALUATION

This section exhibits the simulations to assess the performance of the proposed algorithm in terms of convergence, service acceptance, fairness, and robustness. Drawing from the configuration in [24], [40], our scenario takes shape as a Walker Star LEO satellite constellation, which comprises 100 satellites equally dispersed across 10 LEO orbits positioned 2000 km above the Earth's surface. Additionally, two constellations containing total 25 satellites and 200 satellites are also considered, and realistic location data of satellite Earth stations are set as the locations of each user terminal. The main parameters of our scenario are detailed in Table I. Our evaluation encompasses the performance of our proposed algorithms, named LASO-LAM and GAHSO, juxtaposed against three benchmark approaches. The first benchmark is to deploy VNFs and migrate them in a greedy manner, i.e. the path with minimal hops (MH) and the path with minimal migration numbers (MMN). To examine the performance of LAM, MN with LAM is utilized as the second benchmark. The third benchmark is pheromone-based fault avoidance algorithm (PFA) which is updated from the ant-colony optimization [41]. The simulation is carried out on a computer with 3.0 GHz Intel Core i5-9500, 16 GB RAM, and NVIDIA 2060.

In Fig. 4, we utilize the GCN approach in [34] to evaluate our proposed feature extraction module in convergence. This figure unveils a perceptible enhancement in the performance

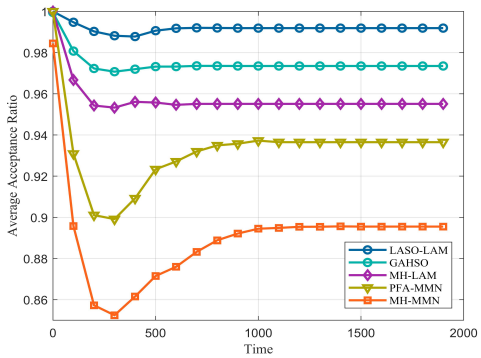

Fig. 5. Comparison of acceptance ratio between proposed algorithms and benchmarks.

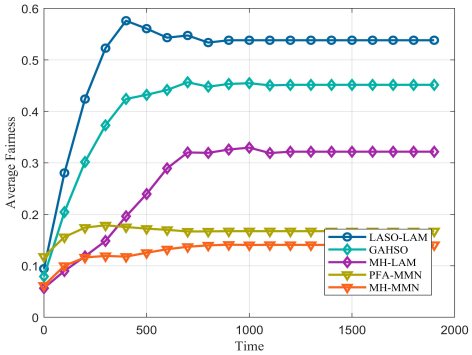

Fig. 6. Comparison of fairness between proposed algorithm and benchmarks.

of both algorithms as the training epoch progresses, ultimately culminating in convergence within approximately 150 epochs. In the training process, it is shown that although the proposed GAT-based RL is weaker than the GCN-based RL at the beginning, it converges to a relatively large value and is more stable than the GCN-based approach after about 50 epochs. This is because the utilized attention mechanism can strengthen the expression of essential information such as available resources in SFC orchestration.

In Figs. 5 and 6, we set the service arrival rate to 3 per slot while maintaining a bandwidth-demand to computation-demand service type ratio of 0.5. Fig. 5 compares the average acceptance ratio of the five algorithms. Notably, LASO-LAM outperforms four benchmarks and achieves nearly 99% of service acceptance. Intriguingly, GAHSO also secures a notable performance, demonstrating merely a 2% decline. The reason is that LASO-LAM and GAHSO utilize the integrated weight of both computation and communication to adapt the multi-dimensional requirements of each SFC, and additional search mechanism is utilized in LASO-LAM to find more potential solutions. Fig. 6 compares the average fairness of the five algorithms, with the LASO-LAM and GAHSO initially appearing lower than PFA-MMN but quickly outperforming all three benchmarks. The reason is the services in PFA avoid passing the nodes in previous solutions. When the number of services increases, LASO-LAM and GAHSO perceive the status of LEO satellite network and schedule the SFCs uniformly. Although the performance of GAHSO is not as good as that of LASO-LAM, GAHSO is also a promising approach considering the simplicity of inference.

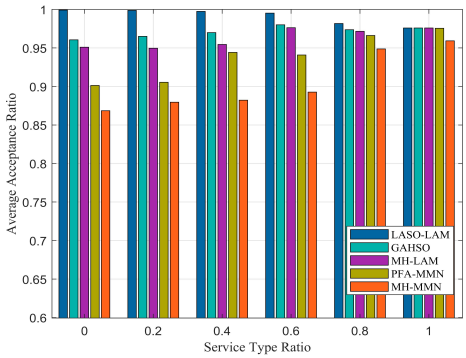

Fig. 7. Service type ratio versus acceptance ratio.

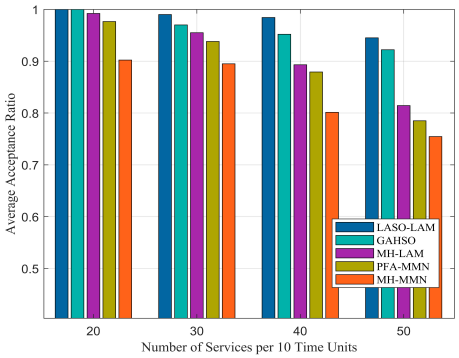

Fig. 9. Arrival rate versus acceptance ratio.

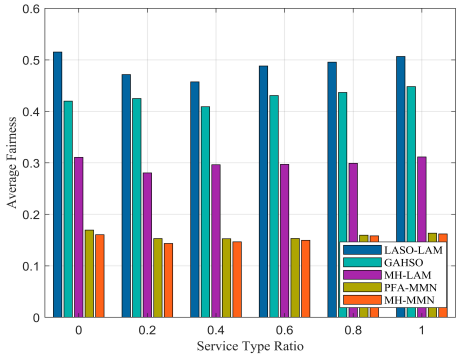

Fig. 8. Service type ratio versus fairness.

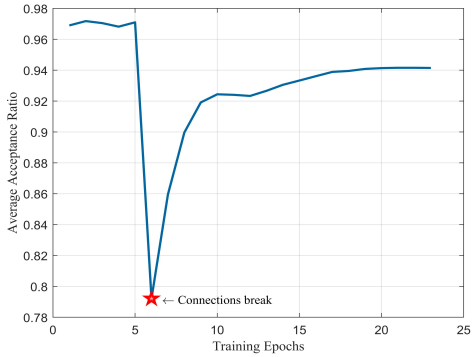

Fig. 10. Acceptance ratio of GAHSO approach under unstable network scenario.

Different types of service influence the status of network in different ways. In Figs. 7 and 8, we show the service acceptance and fairness across diverse ratios of bandwidth-demand to computation-demand services. In Fig. 7, the proposed approaches beat the benchmarks when the ratio is less than 1. This is because our proposed algorithm considers the multi-dimensional network resources and the service requirements concurrently, and too many services with high transmission demands would crowd each other in transmission. Upon scrutinizing ratios lower than 60%, the LASO-LAM exhibits superior performance over GAHSO. This can be attributed to the Dijkstra algorithm's effective evasion of node breakdowns, thereby yielding enhanced outcomes. Similarly, Fig. 8 shows that at the same service ratio, the proposed LASO-LAM achieves twice the fairness value of PFA-MMN and MH-MMN, and nearly 60 percentage points higher than MH-LAM, with the highest service acceptance ratio. The performance of GAHSO is also greater than three benchmarks in service acceptance and load fairness.

Fig. 9 shows the comparison among five algorithms with varied service arrival rates (chosen from [20, 30, 40, 50] per ten time units), and the ratio of each service is set as 0.5. It can be seen from the figure that the proposed LASO-LAM and GAHSO achieve better performance and are more stable under varied service arrival rates. When the service arrival rate is set to 20 per ten time units, the average acceptance ratios of LASO-LAM and GAHSO can approach nearly 100%. With the service arrival rate increasing, the performance of all algorithms decreases, while all benchmarks drop fast. But the GAHSO algorithm can maintain at least 95% of the performance of LASO-LAM algorithm and outperform 20% of the worst approach.

In real LEO satellite networks, the physical environment is dynamic and unstable. Moreover, solar phenomena like coronal mass ejections and solar winds exert disruptive influences, potentially causing disruptions and impairments in satellite communication links [42]. Thus, we simulate to observe the robustness of the trained model under unstable network status where interconnection loss exists in operation. In Fig. 10, we utilize the pretrained model of GAHSO to evaluate its robustness. It is presented that the average acceptance stays around 97% when the network status remains stable. Suddenly at the sixth epoch, random interconnections are broke manually and the average acceptance ratio drops to nearly 79% instantaneously. This is because similar features are extracted when the network remains stable, and the actor network prefers acting similarly. Therefore, some SFCs fail in the sixth epoch. Remarkably, epochs 7 to 10 witness a rapid upswing in performance, culminating in convergence around the twentieth epoch. Compared with the convergence process in Fig. 4 where nearly 150 epochs are required, only after 4 epochs can the pretrained model achieve nearly 95% performance compared with the initial state and nearly 6 more epochs can converge. It is illustrated that the proposed GAHSO approach is suitable for SFC orchestration in dynamic LEO satellite networks, and is able to perceive real-time network topology changes and respond quickly when emergencies occur.

Regarding the practicality of the algorithm, we summarize the average running time for the algorithms to serve 100 SFC requests under three different LEO satellite constellations in Table II. Notably, the novel GAHSO approach shines here,

boasting a consistently stable running time. Even with the satellite network topology expanded from 25 to 200 satellites, the running time merely quadrupled. Comparatively, the LASO-LAM excels in fairness and service acceptance, yet its running time exhibits explosive growth, ballooning nearly 10 times when network size quadruples, and it achieves nearly 31.98 seconds when the number of satellites increases to 200. This phenomenon echoes through MH-LAM and PFA-LAM, owing to their common reliance on the Dijkstra algorithm. As a result, the proposed GAHSO has more potential power in the future large-scale LEO satellite networks for SFC orchestration.

TABLE II
COMPUTATION TIME PER 100 SFCs (SECONDS)

Approach	The Number of Satellites in Network		
	25	100	200
GAHSO	25.265	56.488	98.259
LASO-LAM	66.371	602.150	3197.657
MH-LAM	2.378	129.698	567.932
PFA-LAM	4.792	137.286	749.250

VIII. Conclusion

In this paper, we have investigated the SFC orchestration in LEO satellite networks and proposed an SDN/NFV-based network management architecture to schedule network resources. With the consideration of resource limitations of network infrastructures and service requirements, we have formulated the SFC orchestration as an INLP problem to maximize the service acceptance and the load fairness of the satellite networks. To solve the proposed problems, two approaches with different performance and computation complexity have been presented. Then, multifaceted simulations have been carried out, and the results have demonstrated the effectiveness of the proposed algorithms in terms of service acceptance, fairness, and execution time. Meanwhile, the proposed GAHSO approach has shown robustness when connections between the LEO satellite network are destroyed randomly. The proposed algorithms lay a foundation for further studies in SFC provision and multi-dimensional resource scheduling for LEO satellite networks. In future work, we will leverage the advantage of UAVs in service provision while taking into account the spatio-temporal correlation with the operation of LEO satellite networks.

References

[1] I. Leyva-Mayorga, B. Soret, and P. Popovski, "Inter-Plane Inter-Satellite Connectivity in Dense LEO Constellations," *IEEE T. Wireless Commun.*, vol. 20, no. 6, pp. 3430–3443, 2021.

[2] N. Cheng, W. Xu, W. Shi, Y. Zhou, N. Lu, H. Zhou, and X. Shen, "Air-Ground Integrated Mobile Edge Networks: Architecture, Challenges, and Opportunities," *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 26–32, 2018.

[3] T. Ma, H. Zhou, B. Qian, N. Cheng, X. Shen, X. Chen, and B. Bai, "UAV-LEO Integrated Backbone: A Ubiquitous Data Collection Approach for B5G Internet of Remote Things Networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 11, pp. 3491–3505, 2021.

[4] N. Cheng, F. Lyu, W. Quan, C. Zhou, H. He, W. Shi, and X. Shen, "Space/Aerial-Assisted Computing Offloading for IoT Applications: A Learning-Based Approach," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 1117–1129, 2019.

[5] Z. Yin, N. Cheng, T. H. Luan, and P. Wang, "Physical Layer Security in Cybertwin-Enabled Integrated Satellite-Terrestrial Vehicle Networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 5, pp. 4561–4572, 2022.

[6] J. Li, W. Shi, H. Wu, S. Zhang, and X. Shen, "Cost-Aware Dynamic SFC Mapping and Scheduling in SDN/NFV-Enabled Space-Air-Ground Integrated Networks for Internet of Vehicles," *IEEE IoT J.*, vol. 9, no. 8, pp. 5824–5838, 2021.

[7] O. Bouachir, M. Aloqaily, L. Tseng, and A. Boukerche, "Blockchain and Fog Computing for Cyberphysical Systems: The Case of Smart Industry," *Computer*, vol. 53, no. 9, pp. 36–45, 2020.

[8] H. Peng, D. Li, K. Abboud, H. Zhou, H. Zhao, W. Zhuang, and X. Shen, "Performance Analysis of IEEE 802.11 p DCF for Multiplatooning Communications with Autonomous Vehicles," *IEEE Trans. Veh. Technol.*, vol. 66, no. 3, pp. 2485–2498, 2016.

[9] J. He, N. Cheng, Z. Yin, C. Zhou, H. Zhou, W. Quan, and X.-H. Lin, "Service-Oriented Network Resource Orchestration in Space-Air-Ground Integrated Network," *IEEE Trans. Veh. Technol.*, 2023, early access, DOI: 10.1109/TVT.2023.3301676.

[10] Q. Liu, L. Zeng, M. Bilal, H. Song, X. Liu, Y. Zhang, and X. Cao, "A Multi-Swarm PSO Approach to Large-Scale Task Scheduling in a Sustainable Supply Chain Datacenter," *IEEE Trans. Green Commun. Netw.*, 2023, early access, DOI: 10.1109/TGCN.2023.3283509.

[11] Y. Liu, F. Zhou, S. Member, and T. Shang, "Disaster Protection for Service Function Chain Provisioning in EO-DCNs," *IEEE Trans. Netw. Serv. Manag.*, 2023, early access, DOI: 10.1109/TNSM.2023.3293815.

[12] M. Niu, Q. Han, B. Cheng, M. Wang, Z. Xu, W. Gu, S. Zhang, and J. Chen, "HARS: A High-Available and Resource-Saving Service Function Chain Placement Approach in Data Center Networks," *IEEE Trans. Netw. Serv. Manag.*, vol. 19, no. 2, pp. 829–847, 2022.

[13] Y. C. Wang, R. H. Hwang, and Y. D. Lin, "Low-Latency Service Chaining with Predefined NSH-Based Multipath Across Multiple Datacenters," *IEEE Trans. Netw. Serv. Manag.*, vol. 19, no. 4, pp. 4721–4733, 2022.

[14] M. Wang, B. Cheng, S. Wang, and J. Chen, "Availability and Traffic-Aware Placement of Parallelized SFC in Data Center Networks," *IEEE Trans. Netw. Serv. Manag.*, vol. 18, no. 1, pp. 182–194, 2021.

[15] M. Zhu, J. Gu, T. Shen, J. Zhang, and P. Gu, "Delay-Aware and Resource-Efficient Service Function Chain Mapping in Inter-Datacenter Elastic Optical Networks," *J. Opt. Commun. Netw.*, vol. 14, no. 10, pp. 757–770, 2022.

[16] B. Li and Z. Zhu, "GNN-Based Hierarchical Deep Reinforcement Learning for NFV-Oriented Online Resource Orchestration in Elastic Optical DCIs," *J. Lightwave Technol.*, vol. 40, no. 4, pp. 935–946, 2022.

[17] J. Li, S. Guo, W. Liang, Q. Chen, Z. Xu, W. Xu, and A. Y. Zomaya, "Digital Twin-Assisted, SFC-Enabled Service Provisioning in Mobile Edge Computing," *IEEE Trans. Mobile Comput.*, 2022, early access, DOI: 10.1109/TMC.2022.3227248.

[18] Y. Mao, X. Shang, and Y. Yang, "Joint Resource Management and Flow Scheduling for SFC Deployment in Hybrid Edge-and-Cloud Network," in *Proc. IEEE INFOCOM*, USA, 2022, pp. 170–179.

[19] Y. Hu, M. Chen, W. Saad, H. V. Poor, and S. Cui, "Distributed Multi-Agent Meta Learning for Trajectory Design in Wireless Drone Networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 10, pp. 3177–3192, 2021.

[20] G. Wang, S. Zhou, S. Zhang, Z. Niu, and X. Shen, "SFC-Based Service Provisioning for Reconfigurable Space-Air-Ground Integrated Networks," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 7, pp. 1478–1489, 2020.

[21] P. Zhang, Y. Zhang, N. Kumar, and M. Guizani, "Dynamic SFC Embedding Algorithm Assisted by Federated Learning in Space-Air-Ground Integrated Network Resource Allocation Scenario," *IEEE IoT J.*, vol. 10, no. 11, pp. 9308–9318, 2022.

[22] Z. Chen, N. Cheng, Z. Yin, J. He, and N. Lu, "Service-Oriented Topology Reconfiguration of UAV Networks with Deep Reinforcement Learning," in *Proc. WCSP*, 2022, pp. 753–758.

[23] W. Liu, H. Yang, and J. Li, "Multi-Functional Time Expanded Graph: A Unified Graph Model for Communication, Storage, and Computation for Dynamic Networks Over Time," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 2, pp. 418–431, 2023.

[24] X. Qin, T. Ma, Z. Tang, X. Zhang, H. Zhou, and L. Zhao, "Service-Aware Resource Orchestration in Ultra-Dense LEO Satellite-Terrestrial Integrated 6G: A Service Function Chain Approach," *IEEE T. Wireless Commun.*, vol. 22, no. 9, pp. 6003–6017, 2023.

[25] J. Wang, L. Zhao, J. Liu, and N. Kato, "Smart Resource Allocation for Mobile Edge Computing: A Deep Reinforcement Learning Approach," *IEEE Trans. Emerg. Topics Comput.*, vol. 9, no. 3, pp. 1529–1541, 2021.

[26] M. Tang and V. W. Wong, "Deep Reinforcement Learning for Task Offloading in Mobile Edge Computing Systems," *IEEE Trans. Mobile Comput.*, vol. 21, no. 6, pp. 1985–1997, 2022.
[27] S.-R. Yang, Y.-C. Lin, P. Lin, and Y. Fang, "AIoTtalk: A SIP-Based Service Platform for Heterogeneous Artificial Intelligence of Things Applications," *IEEE Internet Things J.*, vol. 10, no. 16, pp. 14 167–14 181, 2023.
[28] R. Shi, J. Zhang, W. Chu, Q. Bao, X. Jin, C. Gong, Q. Zhu, C. Yu, and S. Rosenberg, "MDP and Machine Learning-Based Cost-Optimization of Dynamic Resource Allocation for Network Function Virtualization," in *Proc. SCC*, USA, 2015, pp. 65–73.
[29] C. Sun, J. Bi, Z. Zheng, and H. Hu, "SLA-NFV: an SLA-aware High Performance Framework for Network Function Virtualization," in *Proc. SIGCOMM*, USA, 2016.
[30] Y. Xiao, Q. Zhang, F. Liu, J. Wang, M. Zhao, Z. Zhang, and J. Zhang, "NFVdeep: Adaptive Online Service Function Chain Deployment with Deep Reinforcement Learning," in *Proc. IWQoS*, USA, 2019.
[31] F. Wei, G. Feng, Y. Sun, Y. Wang, Q. Shuang, and Y.-C. Liang, "Network Slice Reconfiguration by Exploiting Deep Reinforcement Learning With Large Action Space," *IEEE Trans. Netw. Serv. Manag.*, vol. 17, no. 4, pp. 2197–2211, 2020.
[32] Y. Liu, Y. Lu, X. Li, Z. Yao, and D. Zhao, "On Dynamic Service Function Chain Reconfiguration in IoT Networks," *IEEE IoT J.*, vol. 7, no. 11, pp. 10 969–10 984, 2020.
[33] H. G. Kim, S. Park, S. Lange, D. Lee, D. Heo, H. Choi, J. H. Yoo, and J. W. K. Hong, "Graph Neural Network-Based Virtual network function deployment optimization," *Int. J. Netw. Manag.*, vol. 31, no. 6, pp. 2164–2170, 2021.
[34] R. Wang, J. Zhang, Z. Gu, S. Yan, Y. Xiao, and Y. Ji, "Edge-Enhanced Graph Neural Network for DU-CU Placement and Lightpath Provision in X-Haul Networks," *J. Opt. Commun. Netw.*, vol. 14, no. 10, pp. 828–839, 2022.
[35] Y. Xie, L. Huang, Y. Kong, S. Wang, S. Xu, X. Wang, and J. Ren, "Virtualized Network Function Forwarding Graph Placing in SDN and NFV-Enabled IoT Networks: A Graph Neural Network Assisted Deep Reinforcement Learning Method," *IEEE Trans. Netw. Serv. Manag.*, vol. 19, no. 1, pp. 524–537, 2022.
[36] R. Huang, V. W. Wong, and R. Schober, "Throughput Optimization for Grant-Free Multiple Access With Multiagent Deep Reinforcement Learning," *IEEE T. Wireless Commun.*, vol. 20, no. 1, pp. 228–242, 2021.
[37] Z. Xiao, J. Yang, T. Mao, C. Xu, R. Zhang, Z. Han, and X.-G. Xia, "LEO Satellite Access Network (LEO-SAN) towards 6G: Challenges and Approaches," *IEEE Wirel. Commun.*, pp. 1–8, 2022, early access, DOI: 10.1109/MWC.011.2200310.
[38] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous Methods for Deep Reinforcement Learning," in *Proc. ICML*, vol. 48, 2016, pp. 1928–1937.
[39] A. Mohammad, C. Masouros, and Y. Andreopoulos, "Complexity-Scalable Neural-Network-Based MIMO Detection with Learnable Weight Scaling," *IEEE Trans. Commun.*, vol. 68, no. 10, pp. 6101–6113, 2020.
[40] M. Hu, R. Yang, Y. Hu, C. Cai, Y. Dong, T. Deng, and K. Peng, "QoS-Aware Software Defined Multicast in LEO Satellite Networks," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 58, no. 6, 2022.
[41] K. M. Sim and W. H. Sun, "Multiple Ant-Colony Optimization for Network Routing," in *Proc. CW*, Japan, 2002, pp. 277–281.
[42] Z. Yin, N. Cheng, Y. Hui, W. Wang, L. Zhao, K. Aldubaikhy, and A. Alqasir, "Multi-Domain Resource Multiplexing Based Secure Transmission for Satellite-Assisted IoT: AO-SCA Approach," *IEEE T. Wireless Commun.*, 2023, early access, DOI: 10.1109/TWC.2023.3250227.