Optimizing Cost-Efficient SFC Routing in NTNs: A Novel Transformer-Ant Colony Optimization Framework

Yuanfeng Li, *Graduate Student Member, IEEE*, Qi Zhang, *Member, IEEE*, Haipeng Yao, *Senior Member, IEEE*, Xiangjun Xin, Yi Zhao, and Ran Gao

***Abstract*—Non-Terrestrial Networks (NTN), including Low Earth Orbit (LEO) satellite constellations, aim to provide connectivity to remote and underserved areas. These networks now support complex network services leveraging Service Function Chaining (SFC), which sequences service functions such as firewalls and load balances to customize network behavior. In this paper, a comprehensive system model for SFC deployment in NTN is presented. The SFC deployment problem is regarded as comprising two parts: routing and VNF embedding. To improve the cost efficiency of SFC deployment in NTN, a Transformer-Ant Colony Optimization approach has been proposed. The employment of the Transformer architecture to encode SFC requests and the assistance of ACO with neural network modules in identifying optimal paths for VNF embedding are pioneered in this work. The simulation results confirm that our approach achieves superior performance in terms of cost, communication success rate, and delay, demonstrating its potential for efficient SFC deployment in NTN scenarios.**

***Index Terms*—Non-terrestrial networks (NTN), service function chain, ant colony optimization, transformer, low earth orbit satellite.**

Received 2 May 2024; revised 25 August 2024 and 3 November 2024; accepted 29 December 2024. Date of publication 1 January 2025; date of current version 20 May 2025. This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFB2902703, in part by the Funds for Creative Research Groups of China under Grant 62021005, in part by the National Natural Science Foundation of China (NSFC) under Grant 61835002, and in part by the National Natural Science Foundation of China under Grant 62325203 and Grant U22B2033. The review of this article was coordinated by Dr. Zehui Xiong. *(Corresponding authors: Qi Zhang; Haipeng Yao.)*

Yuanfeng Li and Xiangjun Xin are with the School of Electronic Engineering, Beijing University of Posts and Telecommunications (BUPT), Beijing 100876, China, also with Beijing Key Laboratory of Space-ground Interconnection and Convergence, Beijing 100876, China, and also with State Key Laboratory of Information Photonics and Optical Communications, Beijing 100876, China (e-mail: liyuanfeng747@gmail.com; xjxin@bupt.edu.cn).

Qi Zhang is with the School of Electronic Engineering, Beijing University of Posts and Telecommunications (BUPT), Beijing 100876, China, also with Beijing Key Laboratory of Space-ground Interconnection and Convergence, Beijing 100876, China, also with State Key Laboratory of Information Photonics and Optical Communications, Beijing 100876, China, and also with Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha 410073, China (e-mail: zhangqi@bupt.edu.cn).

Haipeng Yao is with the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications (BUPT), Beijing 100876, China (e-mail: yaohaipeng@bupt.edu.cn).

Yi Zhao is with the Beijing Institute of Control and Electronic Technology, Beijing 102300, China (e-mail: 2016010099@bupt.cn).

Ran Gao is with the School of Information and Electronics, Beijing Institution of Technology, Beijing 100081, China (e-mail: gaoran@bit.edu.cn).

Digital Object Identifier 10.1109/TVT.2024.3524583

I. Introduction

In RECENT years, both the academic and industrial sectors have exhibited a marked increase in interest in Non-Terrestrial Networks (NTNs), with a particular focus on Low Earth Orbit (LEO) satellite networks [1], [2], [3]. This attention comes primarily from the unique advantages and potential offered by LEO satellites: they are capable of providing near-ubiquitous network coverage on a global scale, which is particularly valuable for remote and rural areas where terrestrial infrastructure has not yet reached or is prohibitively expensive [4]. As technological advancements continue, these satellites are becoming more cost-effective and seeing improvements in data connection rates and delay performance, making them a viable option to deliver high-speed internet services [5].

As we transition from Beyond 5G (B5G) to 6G, NTNs are increasingly capable of supporting complex network services [6], [7], [8]. Service Function Chaining (SFC), which organizes a series of service functions such as firewalls and load balancers into a specific sequence, is vital for customizing network behavior to diverse use cases [10]. Although LEO satellite constellations inherently contend with challenges such as limited onboard processing capabilities, the expansion of these networks further complicates SFC management due to the growing complexity of the network infrastructure [11]. SFC deployment algorithms for NTNs must navigate these issues, striking a balance between cost efficiency and satellite resource limitations. They must select optimal routing paths to prevent resource exhaustion at VNF activation sites and avoid QoS degradation, while adapting to the dynamism of satellite movement and variable link conditions.

In the previous literature, great research on the optimization of SFC deployment has already been conducted. An approach involves solving the SFC deployment problem through Integer Linear Programming (ILP) techniques [12], [13], [14]. Although this method can yield near-optimal solutions for known SFC requests within polynomial time (i.e. static deployment), it falls short in terms of solution time and flexibility when applied to large-scale NTNs. Another method employs greedy algorithms [15], [16], [17], which calculate SFC routing paths using Dijkstra's or alternative shortest path algorithms. This technique embeds all Virtual Network Functions (VNFs) in a node with the greatest remaining resources along the chosen path. Although this strategy considerably reduces the complexity

of the algorithm, it suffers from a lack of flexibility in VNF embedding, particularly unsuitable for large-scale NTNs where individual nodes have insufficient capacity. Furthermore, there are AI-based algorithms for SFC deployment [18], [19], [20]. The neural combinatorial optimization algorithm for VNF placement treats the network resource distribution and SFC tasks as a black box, necessitating retraining with each new task or resource change, thus lacking flexibility. On the other hand, SFC deployment algorithms based on deep reinforcement learning, despite their flexibility in VNF embedding, are constrained by the limitations of the optimization problem's constraints, making it challenging to converge to the optimal solution.

To address above challenges and improve the cost efficiency for SFC deployment in NTN, a Novel Transformer-Ant Colony Optimization is proposed. The process involves two principal stages: first, the determination of an optimal routing path and second, the embedding of VNFs onto the identified path. Compared with existing researches, the proposed algorithm exhibits superiority by leveraging neural networks to enhance solution quality and stability through learned couplings between resource distributions and heuristic functions. Additionally, Transformer architecture enable optimal node selection for VNF deployment and finer-grained VNF embedding, thus improving resource utilization effectively. The main contributions of this paper can be summarized as follows.

1) In response to the intricate challenge of SFC deployment in NTN, detailed system modeling and problem formulation have been performed, incorporating dynamic network conditions and cost-efficiency considerations, to align with the complex and unstable characteristics of NTN environments.
2) To address SFC deployment challenges in NTN, a hybrid framework has been developed that combines neural network modules with an ant colony optimization algorithm. It introduces a resource-node matching attention mechanism to optimize route discovery by assessing network nodes' affinity with VNFs, improving routing cost efficiency and resource allocation.
3) Using the encoder-decoder structure of the transformer, an innovative VNF embedding mechanism has been introduced for the first time. This mechanism, which adopts an autoregressive approach to VNF embedding suitable for any number of VNFs, significantly improves the utilization of network resources.

The remainder of this paper is organized as follows. In Section II some related works will be introduced. Section III presents our system model and problem formulation. Section IV will introduce our proposed Transformer-ACO SFC routing algorithm. Section V focuses on the simulation results and related analysis. Finally, Section VI summarizes the main conclusions of this paper.

II. Related Work

In addressing the challenge of SFC deployment, the literature reveals a progression from heuristic-based approaches to AI-driven and swarm intelligence-based methodologies.

A. Heuristic-Based Approaches

[21] introduces a framework that utilizes SFC for improved resource management in space-air-ground integrated networks (SAGIN), pioneering the aggregation ratio (AR) metric to optimize the trade-off between communication and computational resources. [22] delves into dynamic VNF mapping and scheduling within SAGINs to boost IoV services, employing Tabu Search heuristics to refine service acceptance rates and provider profit, factoring in migration costs and delays to deliver near-optimal QoS solutions. [23] leverages the Gale-Shapley algorithm to optimize resource allocation in SAGIN, resulting in more efficient task deployment through NFV and SFC.

In addition, [24] examines the SAGIN architecture based on SFC with a focus on delay sensitive applications, advocating a delay prediction methodology for SFC mapping to minimize latency. [25] tackles resource allocation in software-defined LEO satellite networks (SDLSN) by orchestrating VNFs on a software-defined time-evolving graph to reduce the use of S2S links while meeting terrestrial demand, utilizing a branch-and-price algorithm supplemented by approximation and beam search techniques validated through simulations. [26] introduces a novel network architecture to improve service orchestration within the non-terrestrial segments of 6G networks, contending with satellite mobility and terrestrial integration by formulating a dynamic VNF allocation problem based on heuristics for LEO CubeSats. [27] introduces a software-defined network architecture for LEO satellites, employing a MILP-based method and a resource-efficient algorithm for VNF deployment and flow scheduling in large-scale scenarios. [28] proposes a Relative Position-based Efficient Routing (RP-ER) mechanism for LEO satellite networks, enabling low-cost distributed address allocation and redundant routing. By leveraging satellite motion laws, RP-ER reduces resource usage and generates efficient primary and backup routes with concise routing tables. [29] proposes an energy-aware method for multi-domain SFC-enabled networks, introducing a hierarchical control architecture and two algorithms: energy-efficient SFC placement and SFC migration algorithms.

B. AI Driven-Based Approaches

Transitioning to AI-centric strategies, [30] refines the Neural Combinatorial Optimization framework with constraint integration to solve the problem of hard resource placement in the NFV infrastructure, with the goal of reducing power consumption. [31] proposes the sfc2cpu strategy, which merges game theory and neural combinational optimization to effectively map lightweight VNFs onto shared CPU cores within a micro service-based Service Function Chaining framework. [32] introduces DRL-D, a deep reinforcement learning approach for real-time SFC deployment that maximizes long-term revenue within the confines of infrastructure resources, using graph convolution networks and temporal-difference learning for adaptive representation and decision-making of the state of the network. [33] unveils DDQP, an online SFC placement scheme using double deep Q networks to navigate extensive network states and maintain service reliability through active and standby VNF

state synchronizations. [34] offers a service provisioning method using SFC in SAGIN contexts, utilizing NFV for resource allocation and presenting the FL-VNFE algorithm based on federated learning to fine-tune resource distribution with respect to node characteristics and load. [35] introduces a two-stage graph convolution network (GCN)-assisted deep reinforcement learning (DRL) scheme to optimize the embedding of SFC in multi-data center networks, using a macro and micro perspective to handle requests efficiently. [36] proposes a fault-tolerant SFC optimization in SDN/NFV-enabled cloud environments using deep reinforcement learning, addressing challenges of dynamic traffic by enhancing energy efficiency, cost, and load balancing through single and multi-agent approaches. [37] presents an RL-based Q-Learning algorithm for optimizing SFC deployment in edge computing environments, balancing path selection and resource utilization. The approach uses a hierarchical network and a reward function that considers both path efficiency and resource constraints, with evaluations showing effective performance across 1200 test cases.

C. Swarm Intelligence-Based Approaches

Expanding traditional routing problems, the SFC deployment represents an intricate incremental challenge that requires sophisticated solutions [9]. Although Ant Colony Optimization (ACO) algorithms have proven successful in addressing routing issues, their application to SFC deployment, particularly in terms of VNF placement, remains relatively unexplored.

Innovatively, [38] introduces ACO-OSD, an ACO-based meta-heuristic tailored for online SFC deployment, which stands out for its substantial reductions in server operating costs and network latency. Complementing this, the study also presents PLRP, an on-line learning framework that capitalizes on ACO-OSD's strengths to facilitate nearly real time SFC deployment with the advantage of linear time complexity, setting a new precedent for efficiency in the field.

Building further on the potential of ACO, [39] introduces FH-ACO, a novel fuzzy heuristic-enhanced ACO algorithm designed explicitly for the nuanced VNF placement/routing dilemma intrinsic to the NFV landscape. FH-ACO employs a dual multi-criteria fuzzy inference system to guide server/link selection, alongside a robust multi-objective function that judiciously considers power consumption, latency, reliability, and load balancing, showcasing a promising stride toward optimizing complex NFV environments.

III. System Model and Problem Formulation

A. Network Model

The SFC deployment process in a specific NTN scenario is shown in Fig. 1. As the SDN controller for the NTN, the ground station orchestrates network operations from Earth. For task 1, the SFC request is routed into the core network, establishing connectivity that extends to both urban and rural access networks, as well as to ground-based server endpoints. This ensures that users can seamlessly interface with a wide array of services and maintain high-quality communication with various network

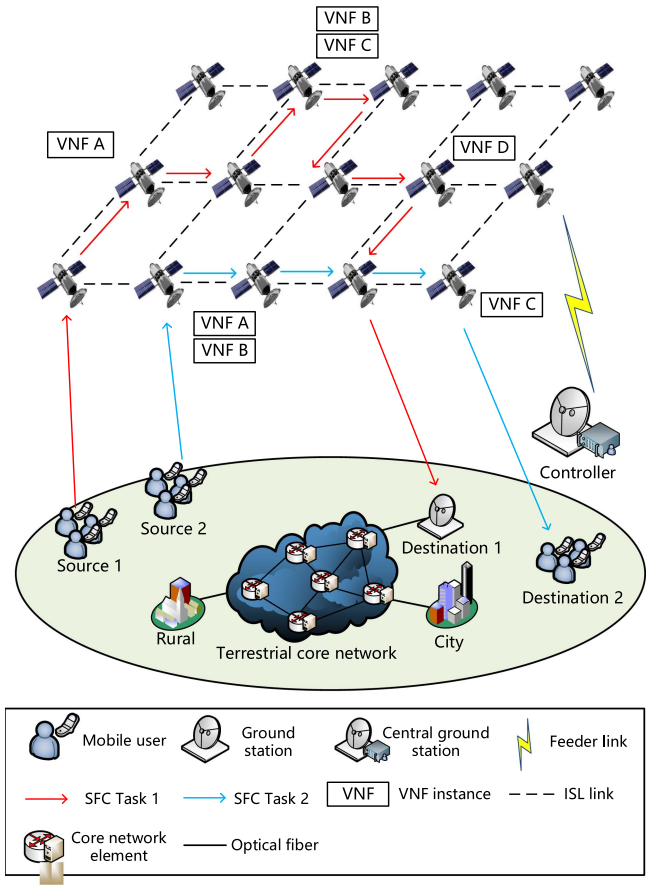


Fig. 1. The scenario of two case of SFC deployment process in NTN.

segments, leveraging the core network's robust infrastructure. Task 2 involves a direct connection from the mobile user to the NTN, enabling users with direct satellite access to transmit their SFC requests without relying on the terrestrial infrastructure. Additionally, the trade-off between these two cases lies in the balance between delay and resource availability. A longer path offers task 1 more opportunities to embed VNFs due to the availability of additional computing and storage resources but introduces higher latency. In contrast, a shorter path reduces task 2 delay but may have fewer resources available, increasing the likelihood of VNF embedding failures.

Consider representing the network as a directed graph $G(V, E)$, where $V = \{V_1, V_2, \ldots, V_S\}$ denotes a set of LEO nodes, with $S = |V|$ being the total number of LEO nodes, and E represents a set of S2S (Satellite-to-Satellite) links. Each link can be denoted as $E_{i,j}$, where i and j are indices that represent nodes V_i and V_j, respectively, which implies that $i, j \in V$.

For each node, we further define the node attributes based on the resource characteristics required by the SFC. For each LEO node V_i, it has a computing resource from the CPU (central processing unit) C_i^{CPU}, RAM (random access memory) resource, C_i^{RAM}, and remaining energy C_i^{Ene}. Similarly, we define the link properties, here each S2S link has delay attribute $\text{Delay}_{i,j}$, bandwidth attribute $\text{B}_{i,j}$, and link duration $\text{Dur}_{i,j}$. The remaining link duration between two satellites, i and j, in adjacent orbits of a Walker-Star constellation and at the same position in their

respective orbits, can be estimated by considering their current orbital positions relative to the polar regions.

Then, we consider a set of SFC requests $\mathcal{R} = \{r_1, r_2, \ldots, r_n\}$, then for each SFC request r, it contains a set of VNF $\mathcal{F} = \{f_{1,r}, f_{2,r}, \ldots, f_{k,r}\}$, the source node S_r and the destination node D_r. In addition, each VNF has its resource demand, computing resource $C^{\text{CPU}}_{f_k,r}$, RAM resource $C^{\text{RAM}}_{f_k,r}$, bandwidth resource $C^{\text{Band}}_{f_k,r}$, energy demand $C^{\text{Ene}}_{f_k,r}$, and request duration time $C^{\text{Dur}}_{f_k,r}$.

To represent the embedding of VNFs onto the network nodes, we introduce a binary variable $x_i^{f_k,r}$, defined as follows:

$$x_i^{f_k,r} = \begin{cases} 1, & \text{the kth VNF } f \text{ of SFC } r \text{ is embedded} \\ & \text{on node } i, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

This variable indicates whether the kth VNF of an SFC request r is hosted on node i ($x_i^{f_k,r} = 1$) or not ($x_i^{f_k,r} = 0$).

Furthermore, the mapping of SFCs on the physical links of the network is captured by another binary variable $y_{i,j}^r$, expressed as:

$$y_{i,j}^r = \begin{cases} 1, & \text{SFC } r \text{ is mapped} \\ & \text{on physical link } E_{i,j} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

This variable determines whether the SFC request r is routed through the physical link between nodes i and j.

Lastly, the success of serving an SFC request r is denoted by the binary variable z_r, which is given by:

$$z_r = \begin{cases} 1, & \text{SFC } r \text{ is successfully served} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

The variable z_r essentially indicates the successful delivery of the SFC request r within the network.

B. S2S Channel Model

The inter-satellite link's bandwidth capacity, denoted by B_{ij}, can be accurately modeled in satellite communication systems through the following equation:

$$\mathrm{B}_{i,j} = \frac{P^{tr} G_{tr} G_{re} L_s L_l}{k_B T_s \left(\frac{E_b}{N_0}\right) m}, \tag{4}$$

where P^{tr} represents the transmit power, G_{tr} is the transmit antenna gain, G_{re} is the receive antenna gain, L_s characterizes the space loss, L_l encompasses other potential losses such as polarization and atmospheric losses, k_B is the Boltzmann constant, T_s is the system noise temperature, $\frac{E_b}{N_0}$ signifies the energy per bit to noise power spectral density ratio, and m stands for the modulation efficiency. The space loss component L_s is further defined as:

$$L_s = \left(\frac{c}{4\pi S f}\right)^2, \tag{5}$$

with c being the speed of light, S the distance between satellites, and f the frequency of the transmitted signal.

C. Energy Consumption Model

The energy consumption model for the SFC nodes is quantified through a series of equations accounting for forwarding, processing, and baseline energy consumption over a time interval Δt. The energy consumption of the forwarding node i is expressed as:

$$E_i^f = \sum_{r\in\mathcal{R}} \left(\sum_{j\in V_s} \frac{P^{tr}\delta_r y_{i,j}^r}{B_{i,j}} + \sum_{j\in V_s} \frac{P^{re}\delta_r y_{j,i}^r}{B_{j,i}} \right) \Delta t, \tag{6}$$

where E_i^f denotes the energy consumed by node i to forward, P^{tr} and P^{re} are the power consumed per unit of data to transmit and receive, respectively, δ_r is the data rate of the service request r, and $B_{i,j}$ represents the bandwidth capacity between nodes i and j.

The processing energy consumption of node i is given by:

$$E_i^V = \sum_{r\in\mathcal{R}} \sum_{f_k\in r} E_c \delta_r \sigma_{f_k} x_i^{f_k,r} \Delta t, \forall i \in V_s, \tag{7}$$

where E_i^V is the energy consumed for processing the service functions within node i, E_c is the energy consumption per unit of processed data, σ_{f_k} is the processing requirement of the service function f_k. It should be noted that for a VNF f_k, its energy demand $C^{\text{Ene}}_{f_k,r}$ is the sum of E_i^f and E_i^V.

Lastly, the baseline energy consumption of node i, which is independent of traffic load, is modeled as:

$$E_i^b = P_b \cdot \Delta t, \tag{8}$$

where E_i^b represents the baseline energy consumed by node i and P_b is the baseline power consumption of the node.

D. Delay Model

The delay model for communication between nodes i and j is defined as the sum of three components: transmission delay $D_{i,j}^{\text{tran}}$, propagation delay $D_{i,j}^{\text{prop}}$, and processing delay D_i^{proc}. This relationship is captured by the following equation.

$$D_{i,j} = D_{i,j}^{\text{tran}} + D_{i,j}^{\text{prop}} + D_i^{\text{proc}}. \tag{9}$$

E. Problem Formulation

1) VNF Placement: When embedding a VNF in a LEO satellite node, it is critical to ensure that the remaining resource capacities of the node are sufficient to meet the requirements of the VNF. The placement constraints can be formulated as follows.

The total CPU requirements for all VNFs on any given node must not surpass the available CPU capacity of that node:

$$\sum_k \sum_r x_i^{f_k,r} \cdot C^{\text{CPU}}_{f_k,r} \cdot z_r \le C_i^{\text{CPU}}. \tag{10}$$

Similarly, the total RAM requirements for all VNFs must remain within the node's available RAM capacity:

$$\sum_k \sum_r x_i^{f_k,r} \cdot C^{\text{RAM}}_{f_k,r} \cdot z_r \le C_i^{\text{RAM}}. \tag{11}$$

The sum of the energy consumption for all active VNFs together with the baseline energy consumption must not exceed the node's energy capacity:

$$\sum_k \sum_r x_i^{f_k,r} \cdot C^{\text{Ene}}_{f_k,r} \cdot C^{\text{Dur}}_{f_k,r} \cdot z_r + E_i^b \cdot T \le C_i^{\text{Ene}}. \quad (12)$$

Each VNF for a given request must be placed on no more than one node:

$$\sum_i x_i^{f_k,r} = 1. \quad (13)$$

2) Flow Routing: Flow routing is governed by the following constraints to ensure that the network's bandwidth and delay requirements are met without exceeding capacities:

For each link between nodes i and j, the sum of bandwidths used by all flows must not exceed the link's bandwidth capacity:

$$\sum_r y_{i,j}^r \cdot C^{\text{Band}}_{f_k,r} \cdot z_r \le B_{i,j}. \quad (14)$$

The duration of use for each link by any flow must be within the link's maximum duration capacity:

$$\forall_r y_{i,j}^r \cdot C^{\text{Dur}}_{f_k,r} \cdot z_r \le \text{Dur}_{i,j}. \quad (15)$$

The total delay for a SFC request r along its path must not exceed the maximum allowable delay for that flow:

$$\sum_{i,j \in \text{path}_r} y_{i,j}^r \cdot D_{i,j} \cdot z_r \le D_{max}^r. \quad (16)$$

Each request r must be routed through exactly one outgoing link from node i:

$$\sum_j y_{i,j}^r = 1. \quad (17)$$

3) Optimization Objective Formulation: The optimization objective for each SFC request r focuses on minimizing the cost associated with routing and VNF embedding while promoting load balancing across the network resources. The cost function is defined as follows:

The routing cost for a request r is determined by the sum of the delays and bandwidth costs over the flow's path:

$$\text{cost}_{r,1} = \sum_{i,j \in \text{path}_r} (D_{i,j} + C^{\text{Band}}_{f_k,r}) \cdot y_{i,j}^r \cdot z_r. \quad (18)$$

The embedding cost for VNFs associated with a request r accounts for the utilization of computational and energy resources after deducting the resources consumed by the VNF embedding, aimed at achieving load balancing.

$$\Delta C^{\text{res}}_{f_k,r} = C^{\text{CPU}}_{f_k,r} + C^{\text{RAM}}_{f_k,r} + C^{\text{Ene}}_{f_k,r} \quad (19a)$$

$$C_i^{\text{res}} = C_i^{\text{CPU}} + C_i^{\text{RAM}} + C_i^{\text{Ene}} \quad (19b)$$

$$\text{cost}_{r,2} = \sum_k \frac{x_i^{f_k,r} \cdot z_r}{C_i^{\text{res}} - \Delta C^{\text{res}}_{f_k,r}}, \quad i \in \text{path}_r \quad (19c)$$

The combined objective is to minimize the total cost, which is a weighted sum of the routing and VNF embedding costs. The weights α and β allow for the adjustment of the relative importance of each component in the objective function:

$$\begin{aligned} \min \quad & \text{cost}_r = \alpha \cdot \text{cost}_{r,1} + \beta \cdot \text{cost}_{r,2} \\ \text{s.t.} \quad & (10)\text{–}(17). \end{aligned} \quad (20)$$

IV. Transformer-Ant Colony Optimization SFC Routing Algorithm

In this section, we first introduce the framework of the proposed algorithm. We will then separately introduce the various neural network modules and the resource-node matching attention mechanism, as well as the training and execution processes of the algorithm.

A. Algorithm Framework

We propose an integrated neural network framework to enhance Ant Colony Optimization (ACO) in finding service function routing paths and making VNF embedding decisions. As shown in Fig. 2, our framework includes five key modules: ACO (yellow), Heuristic Function Generation (green), SFC Encoder (pale blue), VNF Embedding Decision (light purple) and Data Matrix (pink). The ACO Module uses neural heuristic functions to route SFC requests. The Heuristic Function Generation Module employs dual GNN layers and a resource-node matching attention mechanism to generate neural heuristic functions. The SFC Encoder Module leverages Transformer layers to capture VNF relationships in requests. The VNF Embedding Decision Module uses Transformer decoders to place VNFs, updating the path network state after each embedding.

B. Ant Colony Optimization Module

Ant colony optimization searches for routes based on the heuristic function output by the neural network module, which provides routing solutions and resource distribution for the subsequent Transformer module to embed VNFs. In ACO, a colony of ants searches for optimal routing paths using a state transition function. Each ant assesses path costs based on the optimization objective, leaving pheromone trails on paths proportionate to their quality, inversely related to costs. This pheromone feedback adjusts the state transition function in subsequent iterations, directing the ants toward increasingly better solutions.

First, the state transition probability formula is as follows:

$$P(s_t \mid r) = \begin{cases} \frac{\tau_{i,j}^{\alpha} \cdot \eta_{i,j}^{\beta}}{\sum_{l \in \mathcal{N}_i} \tau_{i,l}^{\alpha} \cdot \eta_{i,l}^{\beta}} & \text{if } l \in \mathcal{N}_i, l \notin tabu \\ 0 & \text{otherwise.} \end{cases} \quad (21)$$

The state transition probability $P(s_t|r)$ quantifies the chance of selecting the next node s_t at time t for a SFC request r, in an ant colony optimization process. The selection is based on the level of pheromones $\tau_{i,j}$ and a heuristic desirability $\eta_{i,j}$, modulated by parameters α and β. A transition is possible only if the next node l is a neighbor and not in the $tabu$ list, which avoids cycles.

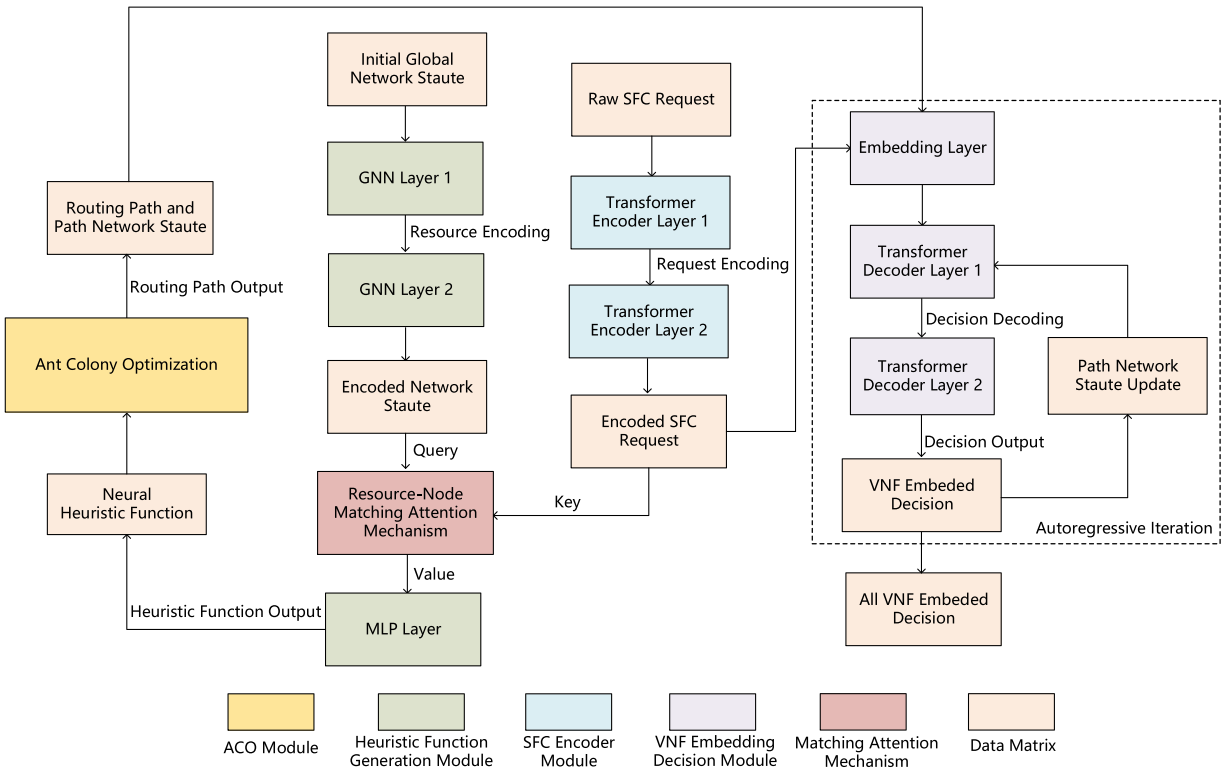


Fig. 2. The framework of Proposed Algorithm.

Then, the state transition probability of a single ant in the complete route search can be expressed as:

$$P(\boldsymbol{s} \mid r) = \prod_{t=1}^{n} P\left(s_t \mid r\right). \tag{22}$$

Since the heuristic function is actually output by the neural network, the state transition probability can be further written as:

$$P_{\boldsymbol{\eta_\theta}}(\boldsymbol{s} \mid r) = \prod_{t=1}^{n} P_{\boldsymbol{\eta_\theta}}\left(s_t \mid r\right). \tag{23}$$

For each ant k that traverses a path, the pheromone levels $\tau_{i,j}$ on the edges of the path are updated as follow:

$$\tau_{i,j} \leftarrow \tau_{i,j} + \frac{1}{\text{cost}_r^n}, \quad \forall (i,j) \in \text{path}_r^n. \tag{24}$$

Here, if path_r^n is the sequence of nodes visited by the nth ant for the SFC request r and cost_r^n is the cost for SFC request r of the nth ant. The pheromone update rule is as follows: for each consecutive pair of nodes (i, j) in path_k, the pheromone level on the edge between nodes i and j increases by the reciprocal of cost_k. This rule applies symmetrically, so that both $\tau_{i,j}$ and $\tau_{j,i}$ are updated, reflecting the undirected nature of the paths.

C. Heuristic Function Generation Module

The heuristic function generation module processes global network statues and encoded SFC requests to produce a specific heuristic value. This heuristic value can guide the initial ant colony to rapidly converge to the optimal solution, which is one of the main advantages of the proposed algorithm.

In the heuristic function generation module, the distribution of network resources and statues in the NTN is represented by a directed graph. The initial global network statues contains node features and edge features. For a given node i, its feature vector $\mathbf{x}_i$ is represented as:

$$\mathbf{x}_i = [C_i^{\text{CPU}}, C_i^{\text{RAM}}, C_i^{\text{Ene}}, \text{Source}_i, \text{Dest}_i], \tag{25}$$

where C_i^{CPU} is the available CPU computation capacity, C_i^{RAM} is the RAM memory resources, C_i^{Ene} indicates the node's energy reserve, Source_i and Dest_i are binary flags identifying whether the node is a source or destination of a SFC request, respectively.

The feature vector for an edge $e_{i,j}$, which connects nodes i and j, is defined as:

$$\mathbf{y}_{i,j} = [B_{i,j}, D_{i,j}, \text{Dur}_{i,j}], \tag{26}$$

where, $B_{i,j}$ represents the available bandwidth resources for the edge between nodes i and j. $D_{i,j}$ denotes the delay associated with the edge. $\text{Dur}_{i,j}$ indicates the remaining duration that the link is expected to be active.

Firstly, we utilize Graph Neural Networks (GNNs) for the initial encoding of the global network statue, which simplifies the complex network interactions into an actionable representation for further processing.

The GNN updates node and edge features across layers using the following rules:

$$\mathbf{x}_i^{l+1} = \mathbf{x}_i^l + \alpha\left(\text{BN}\left(\mathbf{U}^l \mathbf{x}_i^l + \mathcal{A}_{j \in \mathcal{N}_i}\left(\sigma\left(\mathbf{y}_{i,j}^l\right) \odot \mathbf{V}^l \mathbf{x}_j^l\right)\right)\right), \tag{27}$$

where $\mathbf{x}_i^{(l)}$ is the node feature vector of node i at layer l, α is the learning rate, BN denotes batch normalization, $\mathbf{U}^{(l)}$ and $\mathbf{V}^{(l)}$ are weight matrices, σ is a non-linear activation function, $\mathbf{y}_{i,j}^{(l)}$ represents edge features between nodes i and j, $\mathcal{N}(i)$ denotes the set of neighboring nodes of i, and $\odot$ is the element-wise product.

$$\mathbf{y}_{i,j}^{l+1} = \mathbf{y}_{i,j}^{l} + \alpha\left(\mathrm{BN}\left(\mathbf{P}^{l}\mathbf{y}_{ij}^{l} + \mathbf{Q}^{l}\mathbf{x}_{i}^{l} + \mathbf{R}^{l}\mathbf{x}_{j}^{l}\right)\right), \tag{28}$$

where $\mathbf{y}_{i,j}^{l}$ is the edge feature vector of edge (i,j) at layer l, and $\mathbf{P}^{l}$, $\mathbf{Q}^{l}$, and $\mathbf{R}^{l}$ are weight matrices for the edges and the connecting node features. Upon encoding through GNNs, the encoded network state is subjected to a resource-node matching attention mechanism along with the encoded SFC requests, details of which are delineated in the subsequent subsections.

Finally, the data after the resource-node matching attention mechanism is passed through an MLP layer to obtain the neural heuristic function η_{Θ_1}, where $\Theta_1 = \{\theta_{Net}, \theta_{Enc}\}$. θ_{Net} and θ_{Enc} represent the neural network parameters for the heuristic function generation module and the SFC request encoding module, respectively.

To train the heuristic function generation module, we define the following loss function as the optimization objective:

$$\text{minimize} \quad \mathcal{L}\left(\theta_{Net} \mid \boldsymbol{r}\right) = \mathbb{E}_{\substack{\boldsymbol{s}\sim P_{\eta_{\Theta_1}}(\cdot|r) \\ v\sim\mathcal{V}_{\Theta_2}(\cdot|\boldsymbol{s})}}\left[f(\boldsymbol{s}, v) \mid \theta_{Enc}, \boldsymbol{r}\right], \tag{29}$$

here, the loss function is defined as the expected value of the cost function f, which computes the cost based on the route solution $\boldsymbol{s}$ and the VNF embedding solution v. The route solution $\boldsymbol{s}$ is sampled from a distribution $P_{\eta_{\Theta_1}}$ parameterized by Θ_1, while the VNF embedding solution v is sampled from a distribution $\mathcal{V}_{\Theta_2}$ governed by Θ_2, which is a set comprising the encoding parameters θ_{Enc} and the decoding parameters θ_{Dec}. Each part of the equation contributes to determining the overall cost of deploying the SFC in the NTN.

Then, to obtain the gradient of θ_{Net}, we apply a REINFORCE-based gradient estimator:

$$\nabla\mathcal{L}(\theta_{Net}|\boldsymbol{r}) = \mathbb{E}_{\substack{\boldsymbol{s}\sim P_{\eta_{\Theta_1}}(\cdot|\boldsymbol{r}) \\ v\sim\mathcal{V}_{\Theta_2}(\cdot|\boldsymbol{s})}}\left[\left(f(\boldsymbol{s}, v) - b(\boldsymbol{r})\right)\nabla_{\theta_{Net}}\log P_{\eta_{\Theta_1}}(\boldsymbol{s}|\boldsymbol{r})\right], \tag{30}$$

where, $b(\boldsymbol{r})$ represents a baseline function, which is the average cost of SFC solutions searched by ants. It should be noted that although θ_{Net}, θ_{Enc}, and θ_{Dec} are involved in the process of calculating the gradient, we only calculated the gradient of θ_{Net}. This is because we used a pretraining method to train θ_{Enc}, and θ_{Dec} and saved all the parameters.

D. SFC Encoder Module

Encoding a raw SFC request is crucial for capturing the complex dependencies among various VNFs. The Transformer encoder architecture is highly suitable for this task due to its self-attention mechanism, which allows the simultaneous processing of all VNFs and understanding their relationships within the SFC.

We begin by defining the resource requirements for each VNF in the SFC request. The resource vector for the i-th VNF is denoted by $\mathrm{VNF}_i = [C_{f_k,r}^{\mathrm{CPU}}, C_{f_k,r}^{\mathrm{RAM}}, C_{f_k,r}^{\mathrm{Band}}, C_{f_k,r}^{\mathrm{Ene}}, C_{f_k,r}^{\mathrm{Dur}}]$. Consequently, the raw SFC request matrix X, which aggregates the resources for all VNFs, can be formulated as:

$$X = \begin{bmatrix} \begin{pmatrix} C_{f_1,r}^{\mathrm{CPU}} & C_{f_1,r}^{\mathrm{RAM}} & C_{f_1,r}^{\mathrm{Band}} & C_{f_1,r}^{\mathrm{Ene}} & C_{f_1,r}^{\mathrm{Dur}} \end{pmatrix} \\ \begin{pmatrix} C_{f_2,r}^{\mathrm{CPU}} & C_{f_2,r}^{\mathrm{RAM}} & C_{f_2,r}^{\mathrm{Band}} & C_{f_2,r}^{\mathrm{Ene}} & C_{f_2,r}^{\mathrm{Dur}} \end{pmatrix} \\ \vdots \\ \begin{pmatrix} C_{f_n,r}^{\mathrm{CPU}} & C_{f_n,r}^{\mathrm{RAM}} & C_{f_n,r}^{\mathrm{Band}} & C_{f_n,r}^{\mathrm{Ene}} & C_{f_n,r}^{\mathrm{Dur}} \end{pmatrix} \end{bmatrix}$$

The Transformer encoder then utilizes this matrix to generate queries Q, keys K, and values V through linear transformations with the respective weight matrices W^Q, W^K, and W^V:

$$Q = W^Q X \tag{31a}$$

$$K = W^K X \tag{31b}$$

$$V = W^V X. \tag{31c}$$

These are used in the self-attention mechanism, which is the cornerstone of the Transformer model:

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V. \tag{32}$$

The multi-head attention mechanism further refines the model's ability to focus on different positions, where each head captures distinct contextual information:

$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_h)W^O, \tag{33}$$

where W^O is the weight matrix for the linear output transformation, and h represents the number of attention heads. Each head in the multi-head attention component is computed as follows:

$$\mathrm{head}_i = \mathrm{Attention}(W_i^Q Q, W_i^K K, W_i^V V). \tag{34}$$

Beyond attention, each encoder layer includes a position-wise feed forward network, which applies two linear transformations with a ReLU activation in between to each position separately and identically:

$$\mathrm{FFN}(\mathbf{x}) = \max(0, \mathbf{x}W_1 + b_1)W_2 + b_2. \tag{35}$$

Finally, the output of each sublayer, including self-attention and feed-forward networks, is normalized using a layer normalization step. This helps stabilize the learning process and leads to faster convergence:

$$\mathrm{SublayerOutput} = \mathrm{LayerNorm}(\mathbf{x} + \mathrm{Sublayer}(\mathbf{x})). \tag{36}$$

The encoder module described here effectively encodes the SFC request, capturing the nuances and interdependencies of VNF resource requirements within the SFC, which is essential for the subsequent optimization and deployment processes.

E. Resource-Node Matching Attention Mechanism

Although the ant colony may find some routing paths with lower costs, it does not mean that the nodes included in the path have sufficient residual resources suitable for embedding

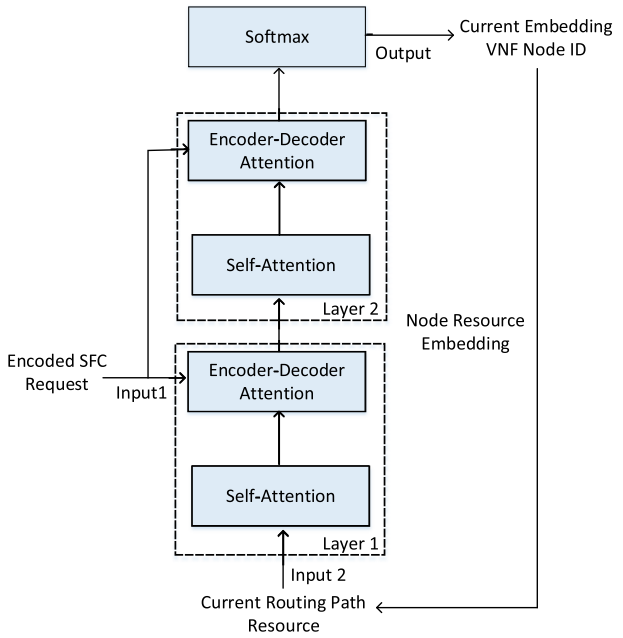


Fig. 3. VNF embedding process based on auto-regression.

VNFs. The resource-node matching attention mechanism assigns higher weights to nodes that better meet the requirements, based on the remaining resources of the nodes and the resource requirements of the SFC.

$$\mathbf{w}_{i,j} = \sum_{k} (\mathbf{y}_{i,j}^{T} \cdot \mathbf{X}_k), \tag{37}$$

here, $\mathbf{X}_k$ represents encoded resources k-th VNF in SFC require.$\mathbf{y}_{i,j}$ represents encoded network statues. A matrix of shape $ij \cdot k$ can be obtained through $\mathbf{y}_{i,j}^{T} \cdot \mathbf{X}_k$, which represents the weighted score of each edge i, j for the k-th VNF embedding. Then we add the k weighted score to obtain the comprehensive weighted score of each edge for the entire SFC request. It should be noted that we represent the node matching information to edges, so the final output is in edge units.

F. VNF Embedding Decision Module

During the VNF embedding process, each VNF embedded consumes the resources of the nodes along the path. The neural network needs to reference the resource changes caused by previous decision-making processes for the next embedding decision. This is very similar to the translation process in NLP, hence we utilize the Transformer architecture to solve this problem.

As depicted in Fig. 3. This figure delineates the architecture of each Transformer decoder layer in conjunction with the autoregressive VNF embedding procedure. The module processes two distinct input streams: the encoded SFC request and the resource data derived from the routing path determined by the ant colony optimization algorithm. Initially, the resource distribution encapsulated within the routing path is encoded by leveraging the self-attention mechanism, with the corresponding equations delineated by (31)–(34).

Subsequently, the encoder-decoder attention mechanism is invoked to assimilate the weighted significance of the encoded SFC request in conjunction with the residual resources of each edge encompassing the routing path. Distinct from the self-attention mechanism, the inputs for the queries (Q), keys (K), and values (V) within the encoder-decoder attention mechanism are delineated as follows:

$$Q = W_q^D H^D \tag{38a}$$

$$K = W_k^E H^E \tag{38b}$$

$$V = W_v^E H^E, \tag{38c}$$

where H^D is the output of the previous layer of the decoder, H^E is the output of the encoder, and W_q^D, W_k^E, W_v^E are the weight matrices for the query, key, and value, respectively. The module ultimately computes the embedding probability of the current VNF at each node along the path via the Softmax function.

It is imperative to underscore that the deployment of all VNFs on the corresponding nodes is executed in an auto-regressive fashion. Post the embedding of each VNF by the decoder, the requisite resources are deducted from the path's resource pool, which then serves as the renewed input to the decoder. This process iterates until the embedding of all VNFs is accomplished.

To train the encoder and decoder, we define the resource distribution variance function V.

$$V = \max\left(0, V_{\text{after}} - V_{\text{ini}}\right) + \alpha \cdot V_{\text{after}}, \tag{39}$$

here, V_{ini} represents the initial resource distribution variance on the routing path, V_{after} represents the resource distribution variance of the path after VNF deployment,and α represents the weighting factor. It can be seen that what affects the a function V is not only the variance of the resources after VNF deployment, but also the difference in variance before and after the deployment.

Based on the above definition of variance function, we give the loss function of θ_{Enc} and θ_{Dec} as follows.

$$\text{minimize}\, \mathcal{L}\left(\theta_{Enc}, \theta_{Dec} \mid \boldsymbol{s}\right) = \mathbb{E}_{v \sim \mathcal{V}_{\Theta_2}(\cdot|s)}\left[V(v \mid \boldsymbol{s})\right]. \tag{40}$$

Here, we obtain the minimized variance function given the routing path solution $\boldsymbol{s}$.

Similarly, we use the REINFORCE-based gradient estimator to obtain the gradients of network parameters.

$$\begin{aligned}\nabla \mathcal{L}\left(\theta_{Enc}, \theta_{Dec} \mid \boldsymbol{s}\right) &= \mathbb{E}_{v \sim \mathcal{V}_{\Theta_2}(\cdot|\boldsymbol{s})}\left[(V(v \mid \boldsymbol{s})\right. \\ &\quad \left. - b(v \mid \boldsymbol{s}))\nabla_{\Theta_2} \log \mathcal{V}_{\Theta_2}(v \mid \boldsymbol{s})\right]. \end{aligned} \tag{41}$$

Here, function $b(v \mid \boldsymbol{s})$ represents the baseline variance function, which uses the Epsilon-Greedy method to obtain the average variance function value under the current network parameters.

G. Training and Implement Process of Algorithm

Firstly, we developed an algorithm to train the SFC encoder and VNF embedding decision modules, detailed in Algorithm1. The input includes an SFC request $\boldsymbol{r}$, an initial routing path $\boldsymbol{s}$ from Original-ACO, and initial parameters Θ_2. The process aims to refine Θ_2' through T_{tr} iterations. During each iteration,

Algorithm 1: Training of SFC encoder module and VNF embedding decision module.

Require: SFC request instance $\boldsymbol{\rho}$, initial routing path solution $\boldsymbol{s}$, initial module parameters Θ_2, number of baseline samples T_b, epsilon-greedy parameter ϵ, number of training iterations T_{tr}

Ensure: Updated module parameters Θ_2'

for iter $= 1$ to T_{tr} **do**
  Acquire routing path solution $\boldsymbol{s}$ for $\boldsymbol{\rho}$ utilizing Original-ACO.
  for iter$_b = 1$ to T_b **do**
    Input raw routing resource distribution and SFC request data to the encoder and decoder, respectively.
    while *Output_num* $<$ *length*(VNFs) **do**
      Determine the node for embedding the current VNF using epsilon-greedy strategy with probability ϵ.
      Update the routing path's resource distribution.
    end while
  end for
  Compute baseline function $b(v \mid \boldsymbol{s})$.
  Evaluate variance function V as per Equation (39).
  Calculate gradients of network parameters following Equation (41).
  Update the module parameters Θ_2 using the computed gradients.
end for

Algorithm 2: Training of Heuristic Function Generation Modules.

Require: SFC request instance $\boldsymbol{r}$, number of ants T_a, initial module parameters θ_{net}, number of training iterations T_{tr}, trained SFC encoder module, and VNF embedding decision module.

Ensure: Updated module parameters θ'_{net}

for iter $= 1$ to T_{tr} **do**
  Acquire SFC request instance $\boldsymbol{r}$.
  Use the trained SFC encoder and heuristic function generation modules to output heuristic information $\boldsymbol{\eta_\theta}$.
  for iter$_{ant} = 1$ to T_a **do**
    while the ant has not reached the Destination Node **do**
      Perform the next-hop state transition according to Equation (21).
    end while
    Generate a solution v using the autoregressive output of the SFC encoder module and VNF embedding decision module.
  end for
  Compute the baseline function $b(r)$.
  Calculate the gradients of network parameters according to Equation (30).
  Update module parameters θ_{net} using the calculated gradients.
end for

the algorithm employs Original-ACO to generate a routing solution for $\boldsymbol{r}$. It then iteratively inputs routing and SFC data into the encoder and decoder. Using an epsilon-greedy approach, it selects nodes for VNF embedding and updates the routing resources. After embedding all VNFs, the algorithm computes the baseline function $b(v \mid \boldsymbol{s})$, assesses the variance function V, and calculates gradients for network parameters. It concludes by updating Θ_2 with these gradients. This training refines the modules for effective SFC and VNF embedding. Then, we present an algorithm for training the heuristic function generation modules in Algorithm 2. For each SFC instance, the virtual ants T_a explore possible solutions, with each ant simulating a path finding process from a source to a destination node. This simulation follows a next-hop state transition defined by Equation (21). The ants' exploration is guided by the heuristic information and the trained modules to autonomously generate potential embedding solutions. Upon completion of the ants' exploration, the algorithm computes a baseline function $b(r)$ used to evaluate the quality of generated solutions. Using this evaluation, the gradients of the network parameters are calculated according to Equation (30). These gradients are then applied to iteratively adjust the parameters of the heuristic function generation modules θ_{net}. The process is repeated for each training iteration, resulting in a continuous refinement of the heuristic function generation modules.

Algorithm 3 describes the execution steps of the algorithm after training. The algorithm initializes by setting pheromone levels and parameters, then processes an SFC request $\boldsymbol{r}$. It operates in multiple colonies T_c, each comprising a number of ants T_a. Within each colony, the ants utilize a trained heuristic function generation module in conjunction with an SFC encoder module to generate heuristic information $\boldsymbol{\eta_\theta}$. Ants traverse from the source to the destination node, with their next-hop transitions determined by combining the heuristic information and current pheromone levels. Concurrently, each ant devises a VNF embedding solution via the autoregressive capabilities of the SFC encoder and the VNF embedding modules. The cost function is applied to evaluate each ant's solution, guiding the pheromone update process to reinforce the more cost-effective paths. After iterating through all colonies and ants, the algorithm identifies the optimal routing solution $\boldsymbol{s}^*$ and VNF embedding solution v^* based on the collective solutions' costs. The Transformer-ACO algorithm's outcome is a set of solutions that are expected to significantly enhance the efficiency of SFC request fulfillment by optimally utilizing network resources.

V. Performance Evaluation

A. Simulation Setting

Our simulation, detailed in Table I. We simulated the Walker constellation using the Matlab 2021 b satellite toolbox. A

TABLE I
DETAILED SYSTEM PARAMETERS

Parameter	Value
Total Bandwidth $B_{i,j}$	2000 Mbps
S2S Delay $D_{i,j}$	[10,15] ms
Bandwidth Demand $C^{\text{Band}}_{f_{k,r}}$	[10,40] Mbps
CPU Demand $C^{\text{CPU}}_{f_{k,r}}$	[0.01,0.05] Tflops
RAM Demand $C^{\text{RAM}}_{f_{k,r}}$	[256,512] MB
Request Arrival Rate	{10,15,20} requests/s
Request Duration $C^{\text{Dur}}_{f_{k,r}}$	[6,10] s
Maximum Delay D^r_{max}	[200,280] ms
CPU Capacity of Node C^{CPU}_i	[1,2] Tflops
RAM Capacity of Node C^{RAM}_i	[512,1024] GB
Energy Capacity of Node C^{Ene}_i	[200,400] W
Model Learning Rate	0.0001

Algorithm 3: Implementation of Transformer-ACO.

Require: SFC request instance $\boldsymbol{\rho}$, number of colonies T_c, number of ants T_a, trained heuristic function generation module, SFC encoder module, and VNF embedding decision module.

Ensure: Optimal routing solution $\boldsymbol{s}^*$ and VNF embedding solution v^*

Initialize pheromone levels and parameters.
Acquire SFC request instance $\boldsymbol{\rho}$.
for iter$_{colony} = 1$ to T_c **do**
  Generate heuristic information $\boldsymbol{\eta_\theta}$ using the trained SFC encoder and heuristic function generation modules.
  for iter$_{ant} = 1$ to T_a **do**
    while the current ant has not reached the destination node **do**
      Determine the next-hop state transition using heuristic information $\boldsymbol{\eta_\theta}$ and pheromone levels, as defined in Equation (21).
    end while
    Construct the VNF embedding solution v_k for the k-th ant using the autoregressive output and VNF embedding decision module.
    Evaluate the cost function $cost_k$ for the k-th ant's solution.
  end for
  Update pheromone levels based on the solutions and their costs, following Equation (24).
  Record the best solution $\boldsymbol{s}_{\text{iter}}$ and v_{iter}.
end for
Select the best solutions $\boldsymbol{s}^*$ and v^* from all iterations based on the cost function.
return $\boldsymbol{s}^*$, v^*

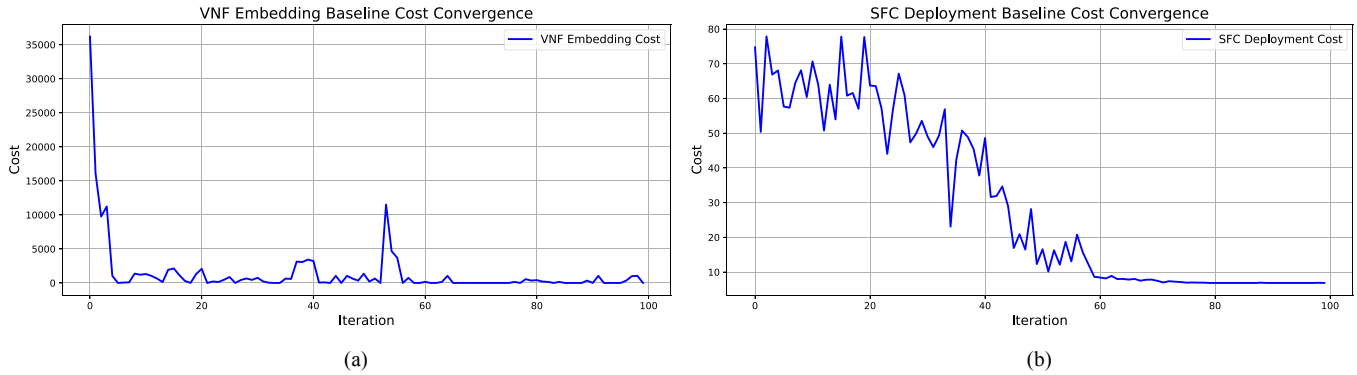
topological snapshot was taken of a single operational period. The resulting snapshot was imported into Python 3.8 using the Networkx library. The experiments were conducted on a device with an Intel i7 13700 k CPU, an RTX4080 GPU, and 32 GB of RAM. PyTorch version 1.12.0 was utilized for the experiments.

B. Convergence Performance

The training phase first trains the SFC request encoding module and the VNF embedding decision module made up of Transformer. We randomly selected the VNF embedding training process requested by a certain SFC. The cost of VNF embedding is the change in the variance in the resource distribution after VNF embedding. The more it costs, the more unbalanced the resource consumption of the node after VNF is embedded. As shown in Fig. 4(a), it shows the baseline cost of VNF embedding, which represents the average expectation of cost incurred by training modules in each iteration. In the early stages of training, the cost of VNF embedding is high, reaching 35000. However, Transformer uses the attention mechanism to perform a strong information representation on path resources and resources requested by SFC. The algorithm converged well within 10 iterations of training, and the lowest cost value reached 0. This shows that the resource distribution variance of the path after the VNF is embedded remains unchanged or becomes smaller, and the resource load is more balanced. Although there are still fluctuations in convergence in subsequent training, this is because we adopt epislon-greedy collection of cost expectations, which leads to random embedding behavior with a certain probability. In the SFC deployment baseline cost, we output the SFC request encoding from the trained Transformer module and deploy the training on the complete SFC request. At this stage, the main training is the graph neural network's ability to represent network resource distribution and its ability to fuse modal information requested by SFC after representation. Although the convergence speed is slower than that of the previous stage of training, it still shows good convergence. As shown in Fig. 4(b), in the early stage of training, its cost is close to 80, and as the number of training rounds continues to increase, the cost gradually decreases. After the number of training rounds reaches 60 rounds, it converges and drops to less than 10.

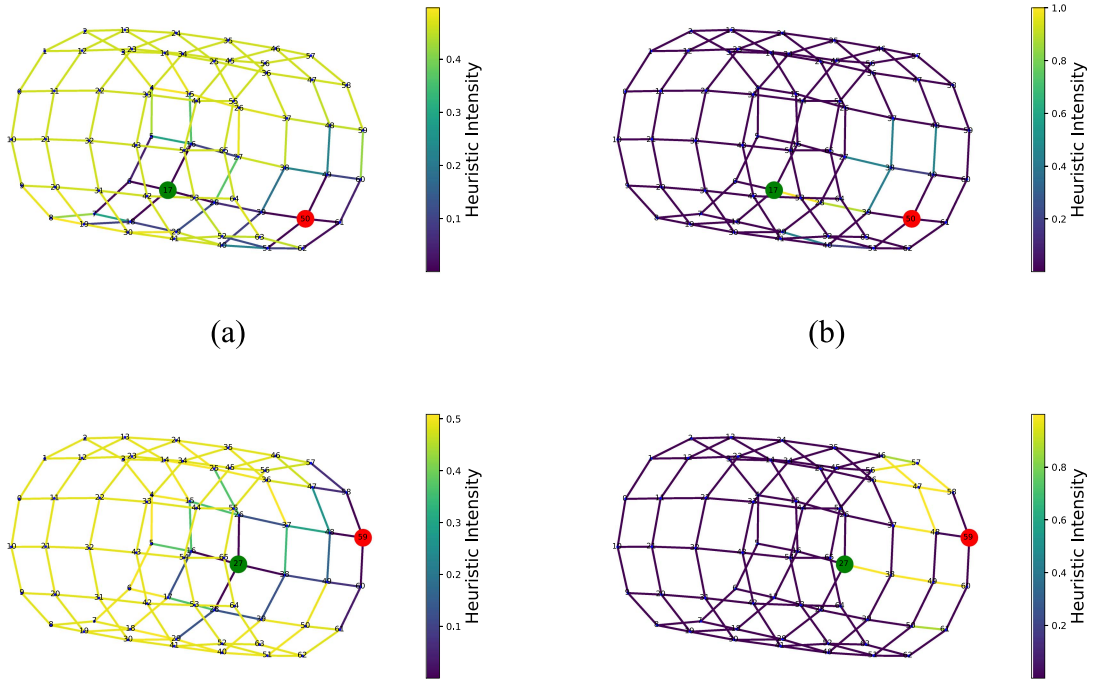
C. Heuristic Function Visualization

In the ant colony algorithm, the heuristic function is an important factor in guiding the ants to find the routes. In order to more intuitively reflect the enhanced effect of our proposed algorithm on ant path finding, we visualize the heuristic function intensity of all link distributions in the DAG of the 66 walker constellation. As shown in Fig. 5, we demonstrate two sets of untrained and trained heuristic function output intensity distribution of different source nodes and destination nodes. Fig. 5(a) and (b) show the untrained and trained heuristic function distributions of source node 17 and destination node 50. In 5(a), the distribution of the untrained heuristic function in the network does not have an obvious directing effect on the ants towards the destination node. The distribution of the heuristic function has no regularity and guides the ants to search further paths. In 5(b), the strength of the heuristic function on the two links, nodes 17 to 28 and nodes 28 to 39, is significantly higher. And the distribution of the heuristic function in other irrelevant areas is basically 0. Similarly, this trend is also shown in 5(c) and 5(d). In 5(c), the distribution of all link heuristic functions is very high and cannot provide effective guidance for ant colony routing, while



(a) (b)

Fig. 4. Baseline cost training convergence of VNF embedding and SFC deployment. (a) VNF embedding baseline cost convergence. (b) SFC deployment baseline cost convergence.



(a) (b) (c) (d)

Fig. 5. Visualized distribution of heuristic function intensity before and after training in 66 walker constellation. (a) Untrained heuristic distribution. (b) Trained heuristic distribution. (c) Untrained heuristic distribution. (d) Trained heuristic distribution.

in 5(d), 27 to 38, 38 to 49, and 49 to 60, the heuristic function has the highest strength. It can effectively guide ants to search for the routing path from the source node to the destination node. By visualizing the heuristic function distribution, we can more intuitively understand why the heuristic function generated by the neural network can guide the ant colony to find a lower-cost route.

D. Cost Performance

We compare the effectiveness of each module in our proposed algorithm and compare the average cost of the algorithm by replacing different modules. We randomly generated 10 s SFC requests and obtained the costs of different ant colonies under different iteration rounds. As shown in Fig. 6, the proposed algorithm without any training does not have any ability to search for SFC solutions. Its cost to search for solutions in all Walker constellations is greater than 50. The maximum cost of other algorithms containing trained modules is around 10. Comparison of the cost of the original ACO combined with Decoder and the proposed algorithm without trained Decoder proves the effectiveness of the proposed SFC request encoder and the heuristic function generation module. Even if the VNF is randomly embedded, the neural heuristic function can still guide the ant colony to search for a better solution than the original ant colony optimization algorithm. As the scale of

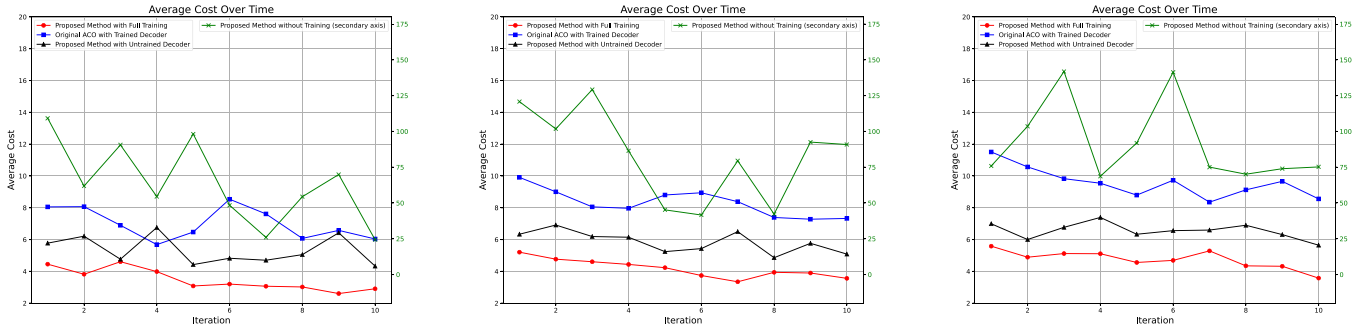


Fig. 6. The average cost of algorithms comparison in different walker constellations. (a) 66 walker constellation. (b) 144 walker constellation. (c) 192 walker constellation.

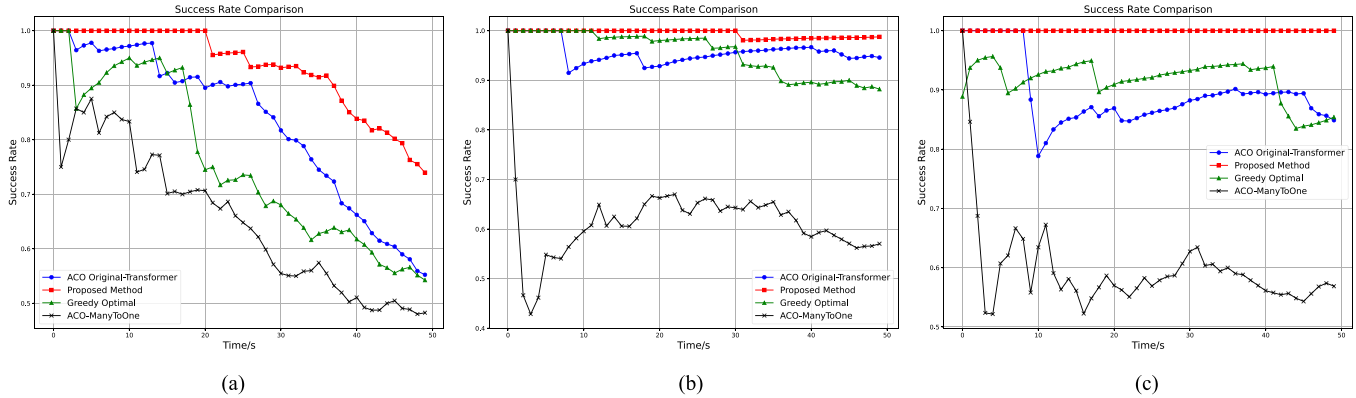


Fig. 7. The communication success rate of algorithms comparison in different walker constellations. (a) 66 walker constellation. (b) 144 walker constellation. (c) 192 walker constellation.

the constellation expands, its effect becomes more significant. Under the 192 walker constellation, compared to the original ant colony combined with the decoder, it can reduce the cost by up to 33 3%. Finally, the fully trained algorithm verifies the effectiveness of the decoder, which can reasonably embed VNF according to the resource distribution of the searched routing path. Compared with the original ant colony combined decoder, it can reduce the cost by up to 54. 5% in the constellation of 192 walker, and compared to the proposed algorithm without training the decoder, it can reduce the cost by up to 28.5%. Through the above comparison of ablation simulation, we can see that our algorithm can effectively optimize SFC requests, and the proposed modules are all effective.

E. Success Rate Performance

We evaluated the communication success rate of the algorithm in the above three walker constellations, randomly generated a series of SFC requests within 50 seconds, and counted the communication success rates of different comparison algorithms. The condition for successful communication is that the ant colony successfully searches for a route from the source node to the destination node and embeds all VNFs under all constraints. The algorithms we compared include two different SOTA algorithms, which are Greedy Optimal [21], ACO-ManyToOne [38], and the Transformer module trained with the original ACO. As shown in Fig. 7, the one with the lowest communication success rate is ACO-ManyToOne. This is because the ACO algorithm mechanism relies on probability transfer to search routing paths. In a large-scale node network, this mechanism is difficult to ensure that the ant colony is searching for the route to the destination node. In addition, the ManyToOne mechanism relies on a greedy algorithm to place VNFs, resulting in a resource load imbalance. After the original ACO is paired with the Transformer, even if the search efficiency of the routing path is not high, the VNF embedding strategy improves the utilization of network resources and effectively improves the communication success rate. The greedy optimal algorithm calculates the route based on the Dijkstra algorithm, which guarantees a routing path from the source node to the destination node. However, due to the lack of flexibility in the VNF embedding mechanism and routing path calculation method, the communication success rate will still decline rapidly. Our proposed algorithm achieves the highest communication success rate among the three scales of satellite network typologies. The heuristic function effectively guides the ant colony in searching for routes and avoids the absence of nodes and links. In addition, more sophisticated VNF

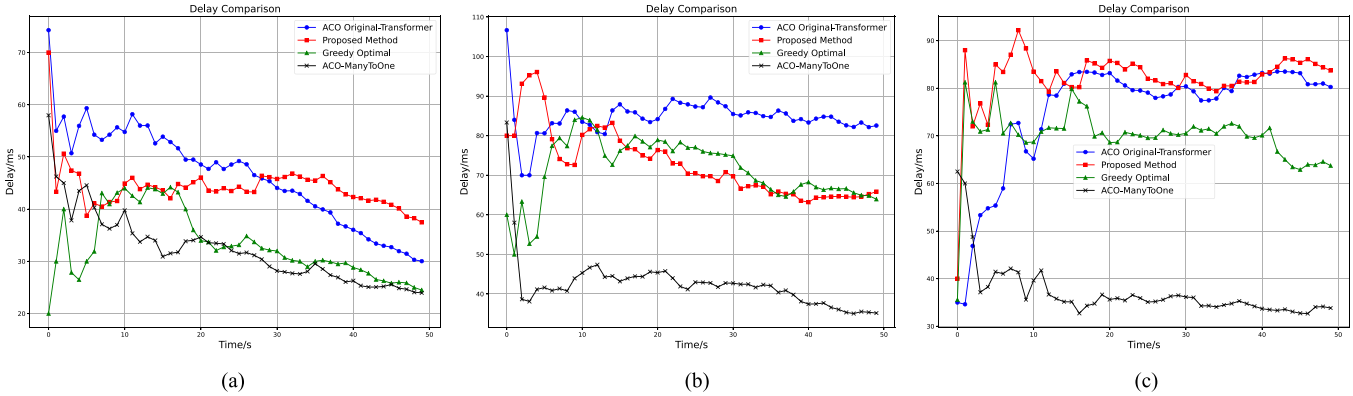


Fig. 8. The communication delay of algorithms comparison in different walker constellations. (a) 66 walker constellation. (b) 144 walker constellation. (c) 192 walker constellation.

deployment strategies improve load balance. The communication success rates in the three Walker constellations have reached 73.5%, 98.3%, and 100%, respectively.

F. Delay Performance

We also collected the communication delays of these comparison algorithms. As shown in Fig. 8, it can be seen that the delay performance of our proposed algorithm is not the lowest. Among these comparison algorithms, the ACO-ManyToOne algorithm has the lowest delay performance. After the 50 s simulation ends, its average delay is less than 40 ms. However, because its communication success rate is the lowest among several algorithms, the reason for this phenomenon is that it lacks the ability to search for routing paths that are far away from the source node and the destination node. Greedy Optimal algorithm shows good communication delay due to the use of Dijkstra algorithm to calculate routing paths. The Greedy Optimal algorithm shows a good communication delay because it uses the Dijkstra algorithm to calculate the route. The final average delay of Greedy Optimal algorithm in the three walker constellations can reach 30 ms, 68 ms, and 63 ms. The algorithm we proposed and the original ant colony with Transformer did not show an advantage in terms of delay. This is because the ant colony must be greater than or equal to the shortest path when looking for long-distance routing paths. Another reason is that the shortest path can cause load imbalance in the network, or the links or nodes in the shortest path do not satisfy resource constraints. The final average delay of our proposed algorithm in the three walker constellations is 38 ms, 67 ms, and 82 ms, respectively.

VI. Conclusion

In this paper, we presented a comprehensive system model for SFC deployment in NTN and proposed a Transformer-ACO based algorithm to address the challenge. We specialized the algorithm for routing and VNF embedding, utilizing a neural networks module tailored to each task. A reinforcement learning strategy informed our baseline cost reduction design. The results confirm that our approach achieves superior performance in terms of cost, communication success rate, and delay, demonstrating its potential for efficient SFC deployment in NTN scenarios. As a reflection on this work, one potential improvement is to use neural networks to fine-tune the heuristic function based on the pheromone distribution in the ACO process, which could enhance solution quality and accelerate convergence. For future work, we will explore large language model (LLM) methods to provide more granular and nuanced routing heuristic strategies tailored to different user policies, thereby improving QoS performance granularity and network robustness.

References

[1] G. Araniti, A. Iera, S. Pizzi, and F. Rinaldi, "Toward 6G non-terrestrial networks," *IEEE Netw.*, vol. 36, no. 1, pp. 113–120, Jan./Feb. 2022, doi: 10.1109/MNET.011.2100191.

[2] M. M. Azari et al., "Evolution of non-terrestrial networks from 5G to 6G: A survey," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 4, pp. 2633–2672, Fourthquarter 2022, doi: 10.1109/COMST.2022.3199901.

[3] M. Harounabadi and T. Heyn, "Toward integration of 6G-NTN to terrestrial mobile networks: Research and standardization aspects," *IEEE Wireless Commun.*, vol. 30, no. 6, pp. 20–26, Dec. 2023, doi: 10.1109/MWC.005.2300207.

[4] H. -L. Maattanen et al., "5G NR communication over GEO or LEO satellite systems: 3GPP RAN higher layer standardization aspects," in *Proc. 2019 IEEE Glob. Commun. Conf.*, Waikoloa, HI, USA, 2019, pp. 1–6, doi: 10.1109/GLOBECOM38437.2019.9014090.

[5] E. Juan, M. Lauridsen, J. Wigard, and P. Mogensen, "Performance evaluation of the 5G NR conditional handover in LEO-based non-terrestrial networks," in *Proc. 2022 IEEE Wireless Commun. Netw. Conf.*, Austin, TX, USA, 2022, pp. 2488–2493, doi: 10.1109/WCNC51071.2022.9771987.

[6] M. Giordani and M. Zorzi, "Non-terrestrial networks in the 6G era: Challenges and opportunities," *IEEE Netw.*, vol. 35, no. 2, pp. 244–251, Mar./Apr. 2021, doi: 10.1109/MNET.011.2000493.

[7] H. -W. Lee, A. Medles, C. -C. Chen, and H. -Y. Wei, "Feasibility and opportunities of terrestrial network and non-terrestrial network spectrum sharing," *IEEE Wireless Commun.*, vol. 30, no. 6, pp. 36–42, Dec. 2023, doi: 10.1109/MWC.001.2300209.

[8] Y. Li, Q. Zhang, H. Yao, R. Gao, X. Xin, and F. R. Yu, "Stigmergy and hierarchical learning for routing optimization in multi-domain collaborative satellite networks," *IEEE J. Sel. Areas Commun.*, vol. 42, no. 5, pp. 1188–1203, May 2024, doi: 10.1109/JSAC.2024.3365878.

[9] Y. Li et al., "Swarm-intelligence-Based routing and wavelength assignment in optical satellite networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 11, no. 1, pp. 1303–1319, Jan./Feb. 2024, doi: 10.1109/TNSE.2023.3321879.

[10] J. Halpern and C. Pignataro, "Service function chaining (SFC) architecture," Internet Eng. Task Force, RFC 7665, Oct. 2015, doi: 10.17487/RFC7665.
[11] Y. Yue et al., "Delay-aware and resource-efficient VNF placement in 6G non-terrestrial networks," in *Proc. 2023 IEEE Wireless Commun. Netw. Conf.*, Glasgow, U.K., 2023, pp. 1–6, doi: 10.1109/WCNC55385.2023.10118893.
[12] D. Li, P. Hong, K. Xue, and J. Pei, "Virtual network function placement considering resource optimization and SFC requests in cloud datacenter," *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 7, pp. 1664–1677, Jul. 2018, doi: 10.1109/TPDS.2018.2802518.
[13] B. Farkiani, B. Bakhshi, and S. A. MirHassani, "A fast near-optimal approach for energy-aware SFC deployment," *IEEE Trans. Netw. Service Manag.*, vol. 16, no. 4, pp. 1360–1373, Dec. 2019, doi: 10.1109/TNSM.2019.2944023.
[14] L. Liu, S. Guo, G. Liu, and Y. Yang, "Joint dynamical VNF placement and SFC routing in NFV-enabled SDNs," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 4, pp. 4263–4276, Dec. 2021, doi: 10.1109/TNSM.2021.3091424.
[15] H. Cao et al., "Resource-ability assisted service function chain embedding and scheduling for 6G networks with virtualization," *IEEE Trans. Veh. Technol.*, vol. 70, no. 4, pp. 3846–3859, Apr. 2021, doi: 10.1109/TVT.2021.3065967.
[16] B. Yuan and B. Ren, "Embedding the minimum cost SFC with end-to-end delay constraint," in *Proc. 5th Int. Conf. Mech., Control Comput. Eng.*, Harbin, China, 2020, pp. 2299–2303, doi: 10.1109/ICMCCE51767.2020.00497.
[17] M. Nguyen, M. Dolati, and M. Ghaderi, "Deadline-aware SFC orchestration under demand uncertainty," *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 4, pp. 2275–2290, Dec. 2020, doi: 10.1109/TNSM.2020.3029749.
[18] Y. Liu, Y. Lu, X. Li, W. Qiao, Z. Li, and D. Zhao, "SFC embedding meets machine learning: Deep reinforcement learning approaches," *IEEE Commun. Lett.*, vol. 25, no. 6, pp. 1926–1930, Jun. 2021, doi: 10.1109/LCOMM.2021.3061991.
[19] S. Long, B. Liu, H. Gao, X. Su, and X. Xu, "Deep reinforcement learning-based SFC deployment scheme for 6G IoT scenario," in *Proc. 2023 IEEE Symp. Comput. Commun.*, Gammarth, Tunisia, 2023, pp. 1189–1192, doi: 10.1109/ISCC58397.2023.10218207.
[20] S. Xu, Y. Li, S. Guo, C. Lei, D. Liu, and X. Qiu, "Cloud–edge collaborative SFC mapping for industrial IoT using deep reinforcement learning," *IEEE Trans. Ind. Informat.*, vol. 18, no. 6, pp. 4158–4168, Jun. 2022, doi: 10.1109/TII.2021.3113875.
[21] 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, Jul. 2020, doi: 10.1109/JSAC.2020.2986851.
[22] 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 Internet Things J.*, vol. 9, no. 8, pp. 5824–5838, Apr. 2022, doi: 10.1109/JIOT.2021.3058250.
[23] Y. Cao, Z. Jia, C. Dong, Y. Wang, J. You, and Q. Wu, "SFC deployment in space-air-ground integrated networks based on matching game," in *Proc. IEEE INFOCOM 2023 - IEEE Conf. Comput. Commun. Workshops*, Hoboken, NJ, USA, 2023, pp. 1–6, doi: 10.1109/INFOCOMWKSHPS57453.2023.10226168.
[24] P. Zhang, P. Yang, N. Kumar, and M. Guizani, "Space-air-Ground integrated network resource allocation based on service function chain," *IEEE Trans. Veh. Technol.*, vol. 71, no. 7, pp. 7730–7738, Jul. 2022, doi: 10.1109/TVT.2022.3165145.
[25] Z. Jia, M. Sheng, J. Li, D. Zhou, and Z. Han, "VNF-Based service provision in software defined LEO satellite networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 9, pp. 6139–6153, Sep. 2021, doi: 10.1109/TWC.2021.3072155.
[26] A. Petrosino, G. Piro, L. A. Grieco, and G. Boggia, "On the optimal deployment of virtual network functions in non-terrestrial segments," *IEEE Trans. Netw. Service Manag.*, vol. 20, no. 4, pp. 4831–4845, Dec. 2023, doi: 10.1109/TNSM.2023.3275248.
[27] M. Xu, M. Jia, Q. Guo, and T. d. Cola, "Delay-sensitive and resource-efficient VNF deployment in satellite-terrestrial networks," *IEEE Trans. Veh. Technol.*, vol. 73, no. 10, pp. 15467–15482, Oct. 2024, doi: 10.1109/TVT.2024.3404090.
[28] Z. Xu et al., "RP-ER: Relative position based efficient routing mechanism for LEO satellite network," in *Proc. GLOBECOM 2023-2023 IEEE Glob. Commun. Conf.*, Kuala Lumpur, Malaysia, 2023, pp. 5967–5972, doi: 10.1109/GLOBECOM54140.2023.10437726.
[29] Q. Xu, D. Gao, H. Zhou, W. Quan, and W. Shi, "An energy-aware method for multi-domain service function chaining," *J. Internet Technol.*, vol. 19, no. 6, pp. 1727–1739, Nov. 2018.
[30] R. Solozabal, J. Ceberio, A. Sanchoyerto, L. Zabala, B. Blanco, and F. Liberal, "Virtual network function placement optimization with deep reinforcement learning," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 2, pp. 292–303, Feb. 2020, doi: 10.1109/JSAC.2019.2959183.
[31] P. Krämer, P. Diederich, C. Krämer, R. Pries, W. Kellerer, and A. Blenk, "sfc2cpu: Operating a service function chain platform with neural combinatorial optimization," in *Proc. 2021 IFIP/IEEE Int. Symp. Integr. Netw. Manage.*, Bordeaux, France, 2021, pp. 196–205.
[32] Q. Fan, P. Pan, X. Li, S. Wang, J. Li, and J. Wen, "DRL-D: Revenue-aware online service function chain deployment via deep reinforcement learning," *IEEE Trans. Netw. Service Manag.*, vol. 19, no. 4, pp. 4531–4545, Dec. 2022, doi: 10.1109/TNSM.2022.3181517.
[33] L. Wang, W. Mao, J. Zhao, and Y. Xu, "DDQP: A. double deep Q-learning approach to online fault-tolerant SFC placement," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 1, pp. 118–132, Mar. 2021, doi: 10.1109/TNSM.2021.3049298.
[34] 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 Internet Things J.*, vol. 10, no. 11, pp. 9308–9318, Jun. 2023, doi: 10.1109/JIOT.2022.3222200.
[35] W. Li, L. Qu, J. Liu, and L. Xie, "Reliability-aware resource allocation for SFC: A column generation-based link protection approach," *IEEE Trans. Netw. Service Manag.*, vol. 21, no. 4, pp. 4583–4597, Aug. 2024, doi: 10.1109/TNSM.2024.3397658.
[36] D. Xiao, J. A. Zhang, X. Liu, Y. Qu, W. Ni, and R. P. Liu, "A two-stage GCN-based deep reinforcement learning framework for SFC embedding in multi-datacenter networks," *IEEE Trans. Netw. Service Manag.*, vol. 20, no. 4, pp. 4297–4312, Dec. 2023, doi: 10.1109/TNSM.2023.3284293.
[37] S. Pandey, J. W., -K. Hong, and J. -H. Yoo, "Q-learning based SFC deployment on edge computing environment," in *Proc. 21st Asia-Pacific Netw. Operations Manage. Symp.*, Daegu, South Korea, 2020, pp. 220–226, doi: 10.23919/APNOMS50412.2020.9236981.
[38] Y. Mao, X. Shang, and Y. Yang, "Ant colony based online learning algorithm for service function chain deployment," in *Proc. IEEE INFOCOM 2023 - IEEE Conf. Comput. Commun.*, New York City, NY, USA, 2023, pp. 1–10, doi: 10.1109/INFOCOM53939.2023.10229012.
[39] M. Shokouhifar, "FH-ACO: Fuzzy heuristic-based ant colony optimization for joint virtual network function placement and routing," *Appl. Soft Comput.*, vol. 107, 2021, Art. no. 107401, doi: 10.1016/j.asoc.2021.107401.

Yuanfeng Li (Graduate Student Member, IEEE) is currently working toward the Ph.D. degree in optical engineering from the School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing, China. His research interests include optical satellite network architecture, network artificial intelligence, multiagent system, space-terrestrial integrated network and network resource allocation.

Qi Zhang (Member, IEEE) received the Ph.D. degree from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2005. She is currently a Professor with the School of Electric Engineering, BUPT. She has authored or coauthored more than 100 SCI papers, in her research field which include optical access network, opticalfiber communication and satellite communication.

Haipeng Yao (Senior Member, IEEE) received the Ph.D. degree from the Department of Telecommunication Engineering, Beijing University of Posts and Telecommunications, Beijing, China, in 2011. He is currently a Professor with the Beijing University of Posts and Telecommunications. He has authored or coauthored more than 150 papers in prestigious peer-reviewed journals and conferences. His research interests include future network architecture, network artificial intelligence, networking, space-terrestrial integrated network, network resource allocation and dedicated networks. Dr. Yao was an Associate Editor for IEEE TRANSACTIONS ON MOBILE COMPUTING and IEEE TRANSACTIONS ON SUSTAINABLE COMPUTING. He was also a member of the technical program committee and Symposium Chair for a number of international conferences, including IWCMC 2019 Symposium Chair and ACM TUR-C SIGSAC2020 Publication Chair.

Xiangjun Xin was born in HeBei, China, in 1969. He received the Ph.D. degree from the School of Electric Engineering, Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2004. He is currently a Professor with the School of Electric Engineering, BUPT. He is a Member of the State Key Laboratory of Information Photonics and Optical Communications, BUPT. He has authored or coauthored more than 100 SCI papers, in his research field which include broadband optical transmission technologies, optical sensor, all-optical network and satellite communication.

Yi Zhao received the Ph.D. degree from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2021. She is currently an Engineer with the Beijing Institute of Control and Electronic Technology, Beijing. Her research focuses on wireless communication, command and control.

Ran Gao received the Ph.D. degree in electronic science and technology from the Beijing Institute of Technology, Beijing, China, in 2015. He is currently an Associate Professor with the Advanced Research Institute of Multidisciplinary Science, Beijing Institute of Technology. His research interests include fiber optical sensors, optical communication, and measurement instruments.