A Two-Stage Heuristic SFC Deployment Approach in Software Defined Satellite Networks

Zhanjun Liu, Feng Peng, Chengchao Liang(✉), and Rong Chai

School of Communications and Information Engineering,
Chongqing University of Posts and Telecommunications, Chongqing, China
`{liuzj,liangcc,chairong}@cqupt.edu.cn`

Abstract. Integrating network function virtualization (NFV) into service function chaining (SFC) allows for the flexible processing and forwarding of traffic along predetermined virtual network functions (VNFs), thus effectively coordinating resource allocation in software-defined satellite networks (SDSNs). However, the deployment of SFC in satellite networks is more complex than in terrestrial networks due to certain characteristics of satellites, such as dynamic topology and limited payloads. This paper proposes a delay minimization problem that couples VNF deployment and routing, and then presents a two-stage heuristic algorithm to solve it. The first stage employs an improved particle swarm optimization algorithm, while the second stage is a traffic routing algorithm based on Time-Evolution Graph (TEG). Finally, simulations are conducted under various system settings to validate the effectiveness of the algorithm.

Keywords: Service function chain · software defined satellite networks · particle swarm optimization · traffic routing

1 Introduction

Satellite networks have been a major factor in the development of communication technology due to their ability to provide global coverage and low latency. In areas where ground networks are difficult to access, satellite networks provide a vital means of communication, allowing users in these regions to benefit from high-quality services [1]. However, traditional satellite network structures are not very flexible and cannot meet the diverse and personalized network service requirements. This is why the introduction of Network Function Virtualization (NFV) and Software Defined Networking (SDN) technologies is so important [2]. NFV allows for dynamic deployment and scheduling in a virtual environment by separating network functions from hardware devices [3]. SDN, on the other hand, separates the control layer from the data forwarding layer, giving network administrators more control over network traffic and allowing them to meet complex and ever-changing network service demands [4].

F. Gao et al. (Eds.): Chinacom 2023, LNICST 590, pp. 36–49, 2024.
https://doi.org/10.1007/978-3-031-67162-3_3

The implementation of a satellite network with NFV and SDN technologies presents a challenge in terms of coordinating and optimizing resource allocation. Service Function Chain (SFC) has been identified as an effective solution for service provision. This scheme defines a set of ordered Virtual Network Functions (VNFs) and guides the requested data traffic flow through these VNFs in a specified order. By doing so, SFC can make use of the embedded computing resources in the network and flexibly forward and process user traffic between network service nodes deploying different VNFs through the collaborative allocation of network resources and computing resources [5,6].

The issues related to the deployment of SFCs involve mainly the basic problems of placing VNFs requested by each user and determining the routes between each adjacent pair of VNFs without exceeding the capacity of the nodes and the bandwidth of the links [7]. This can be seen as a generalization of the well-known Virtual Network Embedding (VNE) problem, which has been proven to be NP-hard. Consequently, heuristic algorithms are commonly used to address SFC deployment issues, particularly those based on Integer Linear Programming (ILP) and Mixed Integer Linear Programming (MILP) approaches [8]. Additionally, game theory, graph optimization, approximation, and evolutionary algorithms have also been employed to solve VNF placement problems.

Gao et al. [9] investigated the SFC deployment problem in a satellite ground station network and proposed a location-aware resource allocation algorithm based on a greedy approach. Cai [10] explored the deployment of SFCs in satellite communication networks and developed an effective heuristic algorithm to reduce the total end-to-end delay and balance the load. Wang [11] proposed a reconfigurable service provision framework for SFCs and a heuristic greedy algorithm to address the SFC planning problem. Yang et al. [12] suggested a genetic-based VNF deployment algorithm to match SFCs with available system resources.

Despite the considerable amount of research devoted to the SFC deployment problem, most studies have overlooked the dynamic topological characteristics of satellite networks. The movement of satellites and the instability of communication links cause the network topology to vary from one time slot to the next, making routing planning for SFC more complex. Fortunately, [13] proposed a heuristic slot decoupling algorithm that optimizes VNF deployment and routing simultaneously, resulting in considerable cost savings. [14] investigated the deployment of dependable functional service chains with the help of backup and proposed a heuristic algorithm based on TEG, which reduces the total upper limit delay.

Nevertheless, the progress mentioned above does not consider that when multiple slots and SFCs are deployed, different holding times will have an effect on the total service delay. Time delay, which is a significant measure of SFC performance, is a very crucial factor in SFC deployment. Therefore, the main contributions of this paper are:

- Taking into account the lingering delay and the limited satellite payload, we established the detailed model of SFC deployment on SDSN, which is a novel and innovative work focusing on optimizing uncertain delay in satellite networks.

- A two-stage heuristic approach is suggested, beginning with an improved particle swarm optimization (IPSO) algorithm, followed by a traffic routing algorithm based on TEG.
- In the simulation phase, the performance of IPSO is compared with a variety of heuristic algorithms, and then the proposed algorithm's effectiveness is tested in resolving the SFC deployment problem in SDSN.

This paper is structured in the following way. Section 2 presents the system model and problem modeling. Section 3 outlines the two-stage heuristic algorithm proposed. Section 4 provides the simulation results. Finally, Sect. 5 offers the conclusions.

2 System Model and Problem Formulation

2.1 Network Model

As depicted in Fig. 1, the Network Operations Control Center (NOCC) and GEO satellites constitute the control plane of the SDSN network, while the LEO satellite network serves as the data plane. To account for the intermittent and predictable characteristics of inter-satellite links, TEG is used to represent the various resources in the satellite network. This two-dimensional topology graph, composed of time and space, is capable of describing the periodic and dynamically changing nature of the satellite network. It functions as a physical substrate network for deploying SFCs, thus allowing for the effective management and utilization of the various resources in the SDSN [15].

This paper divides the total time into T intervals (slots), each lasting τ. It is assumed that the network topology remains constant during each time period $t \in T$, and that rapid changes occur when the time intervals switch.

The TEG can be represented as $G(N, E)$, where N is the set of nodes and E is the set of links. In the TEG, there are two different types of links: inter-satellite links (i^t, j^t) between two distinct satellites and storage arcs (i^t, i^{t+1}) between successive time slots of the same satellite. We assume that the storage capacity of a LEO satellite is infinite, while its computing resource capacity is finite. The computing resource capacity of node $i \in N$ is denoted by C_i^t.

The bandwidth resource capacity of the inter-satellite links (i^t, j^t) between different satellites is represented by B_{ij}^t. If there is no inter-satellite link between i^t and j^t, it is denoted as $B_{ij}^t = 0$. The node set includes ground source nodes N_s, satellite nodes N_v, and ground destination nodes N_d. The link set includes the uplink set from the ground to satellites E_{sv}, the inter-satellite link set E_{vv}, the downlink set from satellites to ground E_{vd}, and the set of satellite's storage arcs E_v in adjacent time slots.

2.2 SFC Request Model

The SFC request of the user k is composed of a set of VNFs arranged in a specific order. It is represented by the tuple $< S_k, D_k, F_k, d_k >$, where S_k and D_k are

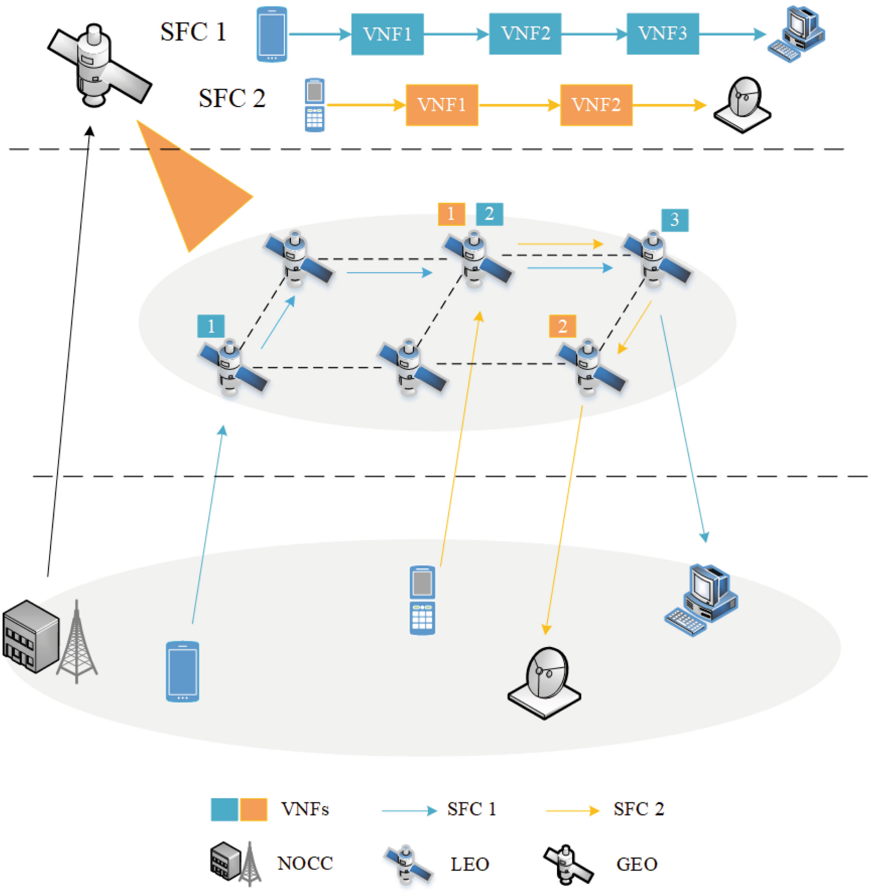

Fig. 1. Software Defined Satellite Network

the source and destination nodes, respectively. $F_k = (f_1^k, f_2^k, f_3^k, ..., f_n^k)$ is the ordered set of VNFs, and d_k is the data size. Furthermore, c_n^k is the computing resource demand of the nth VNF in the SFC request, while b_n^k is the bandwidth demand of the virtual link from f_n^k to f_{n+1}^k. The satellite network can provide multiple types of VNFs.

As illustrated in Fig. 2, we use the deployment of SFC 1 as an example. The blue dotted line and the blue solid line represent two different paths. Path 1 is $1^1 \rightarrow 4^1 \rightarrow 5^1 \rightarrow 5^2 \rightarrow 6^2$ and path 2 is $1^1 \rightarrow 4^1 \rightarrow 5^1 \rightarrow 2^1 \rightarrow 3^1 \rightarrow 6^1$. In a static topology, path 2 is the only route to consider. However, due to the dynamic topology of the satellite, path 1 is now an option. Using the storage arc link, the traffic is temporarily stored in 5^1 until the next time slot, and then is transmitted from $5^2 \rightarrow 6^2$. Since the optimization of delay is of great importance and path 1 consumes fewer bandwidth resources, it is beneficial to analyze the differences between the two paths. However, the time delay consumed by storage

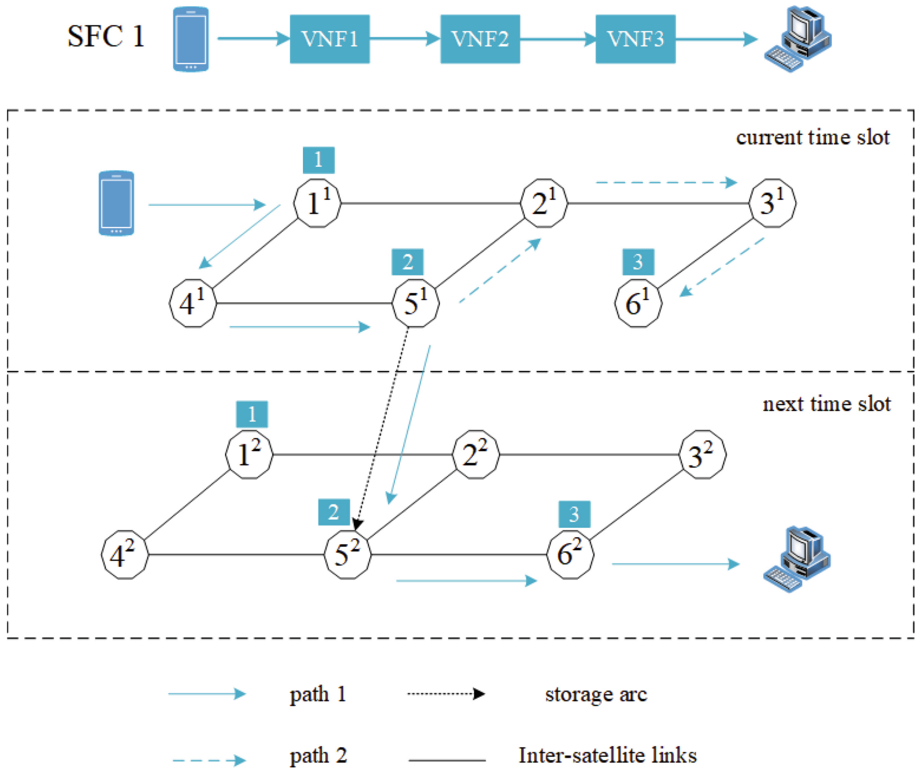

Fig. 2. An illustration of SFC deployment on TEG (Color figure online)

arcs in path 1 is not fixed, so an effective SFC deployment strategy is needed to guide the routing of traffic between the VNF deployment nodes.

2.3 Problem Formulation

The deployment of VNF in the satellite network is described as $x_n^{i^t}$, where $x_n^{i^t} = 1$ indicates that VNF f_n^k is deployed on satellite i^t, and $x_n^{i^t} = 0$ means it is not deployed. We assume that VNF f_n^k can only be deployed on one satellite node, so we have:

$$\sum_{i^t \in N} x_n^{i^t} = 1, \quad \forall k \in K, f_n^k \in F_k \tag{1}$$

The nth virtual link l_n^k of SFC request k is considered the nth service stage of the SFC. A binary decision variable $u_n^{i^t, j^d}$ is used to indicate the relationship between link (i^t, j^d) and the nth SFC service stage. If link (i^t, j^t) is used for the nth stage, $u_n^{i^t, j^t} = 1$, otherwise $u_n^{i^t, j^t} = 0$. For the deployment of the storage arc, $u_n^{i^t, i^{t+1}} = 1$ implies that the storage arc (i^t, i^{t+1}) is employed in the nth stage, otherwise it is 0.

The traffic engineering theory states that when $x_n^{i^t} = 1$, node i^t can be considered the source node of the nth stage of SFC. As a result, the difference between the outgoing and incoming traffic for a source node is equal to 1, which can be expressed as:

$$\sum_{(i^t,j^d)\in E} u_n^{i^t,j^d} - \sum_{(j^d,i^t)\in E} u_n^{j^d,i^t} = x_n^{i^t} \tag{2}$$

When $x_{n+1}^{i^t} = 1$, node i^t is the destination node of the n^{th} stage of the SFC. The difference between the outgoing and incoming traffic for a destination node is -1, thus:

$$\sum_{(i^t,j^d)\in E} u_n^{i^t,j^d} - \sum_{(j^d,i^t)\in E} u_n^{j^d,i^t} = -x_{n+1}^{i^t} \tag{3}$$

For forwarding nodes (traffic nodes without VNF deployed) and non-traffic nodes, outgoing traffic and incoming traffic are conserved:

$$\sum_{(i^t,j^d)\in E} u_n^{i^t,j^d} - \sum_{(j^d,i^t)\in E} u_n^{j^d,i^t} = 0 \tag{4}$$

In fact, each service node may deploy multiple VNFs of the same SFC. If both VNF f_n^k and f_{n+1}^k are deployed on node i^t, then the traffic transmission is carried out inside the node, so the incoming traffic and outgoing traffic are both 0:

$$0 = x_n^{i^t} - x_{n+1}^{i^t} \tag{5}$$

Integrating (2), (3), (4), and (5), the flow constraints of the TEG can be obtained as:

$$\sum_{(i^t,j^d)\in E} u_n^{i^t,j^d} - \sum_{(j^d,i^t)\in E} u_n^{j^d,i^t} = x_n^{i^t} - x_{n+1}^{i^t}, \forall i^t \in N,\ \forall k, n \tag{6}$$

For resource constraints, each service node and each inter-satellite link has corresponding computing resource capacity and bandwidth resource capacity.

$$\sum_k \sum_n c_n^k x_n^{i^t} \leq C_i^t, \quad \forall i^t \in N \tag{7}$$

$$\sum_k \sum_n b_n^k u_n^{i^t,j^t} \leq B_{ij}^t, \quad \forall (i^t, j^t) \in E_{vv} \tag{8}$$

Assuming that the service delay of the nth stage of the kth SFC is $\delta_{k,n}^d$, it includes the calculation delay $\delta_{k,n}^c$ of VNF f_n^k on the relevant node, and the transmission delay $\delta_{k,n}^t$ of SFC traffic from the deployment node of VNF f_n^k to the deployment node of VNF f_{n+1}^k and storage arc delay $\delta_{k,n}^s$. The transmission delay of the inter-satellite link can be expressed as:

$$\delta_{k,n}^t = \sum_{(i^t,j^t)\in E_{vv}} \delta_{k,n}^l u_n^{i^t,j^t} \tag{9}$$

where $\delta_{k,n}^{l} = d_k/b_n^k$, d_k denotes the total amount of data of the kth SFC, and b_n^k represents the transmission rate at the current stage.

Therefore, without using storage arc links, the service delay of the nth service stage of SFC k can be expressed as:

$$\delta_{k,n}^{d} = \delta_{k,n}^{c} + \sum_{(i^t,j^t)\in E_{\nu\nu}} \delta_{k,n}^{l} u_n^{i^t,j^t} \tag{10}$$

where $\delta_{k,n}^{c} = d_k/c_n^k$, and c_n^k is the computing resource rate at the current stage. In addition to calculation delay and link transmission delay, storage delay may occur due to the dynamic nature of satellite topology. Assuming that each SFC will only use the storage arc at most once, when we use the storage arc link in the nth service phase, then the total time delay consumed before the storage arc transmission is:

$$\delta_{total}^{d} = \sum_{n-1}^{i} \delta_{k,i}^{d} + \delta_{k,n}^{c} \tag{11}$$

When $\delta_{total}^{d} < \tau$, there will be a storage delay, and the storage delay is:

$$\delta_{k,n}^{\mathrm{s}} = \sum_{(i^t,i^{t+1})\in E_v} (\tau - \delta_{total}^{d}) u_n^{i^t,i^{t+1}} \tag{12}$$

where τ is the duration of a time slot. Therefore, in the case of using storage arc links, the service delay of the nth service phase of SFC k is:

$$\delta_{k,n}^{d} = \delta_{k,n}^{c} + \sum_{(i^t,j^t)\in E_{vv}} \delta_{k,n}^{l} u_n^{i^t,j^t} + \delta_{k,n}^{\mathrm{s}} \tag{13}$$

In the case of on-demand allocation of node computing resources and link transmission resources required for each service phase of SFC, we should also meet the maximum tolerable delay requirement, so:

$$\sum_{n} \delta_{k,n}^{d} \le \delta_{k}^{m}, \quad \forall k, n \tag{14}$$

All of the above constitute our optimization problem, and the optimization problem can be expressed as:

$$
\begin{aligned}
&\textbf{P0:} \min_{X,U} \sum_{k} \sum_{n} \delta_{k,n}^{d} \\
&s.t.\, C1\text{:}\ \sum_{i^t \in N} x_n^{i^t} = 1, \quad \forall k, n \\
&\quad C2\text{:}\ \sum_{(i^t, j^d) \in E} u_n^{i^t, j^d} - \sum_{(j^d, i^t) \in E} u_n^{j^d, i^t} = x_n^{i^t} - x_{n+1}^{i^t}, \quad \forall i^t \in N, \quad \forall k, n \\
&\quad C3\text{:}\ \sum_{k} \sum_{n} c_n^k x_n^{i^t} \leq C_i^t, \quad \forall i^t \in N \\
&\quad C4\text{:}\ \sum_{k} \sum_{n} b_n^k u_n^{i^t, j^t} \leq B_{ij}^t, \quad \forall (i^t, j^t) \in E_{vv} \\
&\quad C5\text{:}\ \sum_{n} \delta_{k,n}^d \leq \delta_k^m, \quad \forall k, n \\
&\quad C6\text{:}\ x_n^{i^t} \in \{0, 1\}, \quad \forall i^t \in N, \quad \forall k, n \\
&\quad C7\text{:}\ u_n^{i^t, j^d} \in \{0, 1\}, \quad \forall (i^t, j^d) \in E_{vv} \cup E_v, \quad \forall k, n
\end{aligned} \tag{15}
$$

The objective of the optimization problem is denoted as **P0**. To ensure that VNF x is only deployed on one satellite node, constraint $C1$ is imposed. Flow conservation constraint $C2$ is applied to guarantee that SFC traffic starts from the source node and reaches the destination node through the VNF deployment nodes in the correct order. Constraints $C3$ and $C4$ are related to the computing resources of the nodes and the bandwidth of the links, respectively. Maximum tolerable delay constraint $C5$ is also included. Binary variables $C6$ and $C7$ are used for node selection and link selection, respectively.

3 Two-Stage Heuristic Algorithm

Aiming at the SFC deployment problem of SDSN network discussed in this paper, we propose a two-stage heuristic SFC deployment algorithm (THSDA) to solve this problem. For the VNF deployment problem, we use an improved PSO algorithm (IPSO) to solve this problem, considering the fact that the PSO algorithm is easy to fall into local optimum; for the SFC routing problem, considering the instability of the satellite topology, a routing algorithm based on delay sensitivity is adopted.

3.1 IPSO Algorithm

Traditional Particle Swarm Optimization (PSO) is only applicable to continuous problems, whereas the Virtual Network Function Placement (VNFP) problem is a discrete one. Binary coding is suitable for general VNFP problems, but it cannot be used to route according to the sequence predetermined by the Service

Function Chaining (SFC). Therefore, we use the serial numbering method to address the VNF deployment issue.

First, we assign an integer ID to each satellite node, assuming that the ordered VNF node set $F_k = (f_1^k, f_2^k, f_3^k, ..., f_n^k)$ of the SFC k corresponds to the VNF sequence $Seq_k = (v_1^k, v_2^k, v_3^k, ..., v_n^k)$, and v_n^k represents the ID of deployed nodes. Then, based on the VNF sequence Seq, construct an SFC path $Path(Seq)$, which is the SFC routing problem. The details will be given in the next section, but the fitness value needs to be calculated by $Path(Seq)$ to calculate the SFC delay. Because our optimization goal is the total delay of multiple SFCs, the solution X should be defined as a two-dimensional array, that is, $X = (Seq_1, Seq_2, Seq_3, .., Seq_k)^T$, where Seq_k represents the VNF sequence of the kth SFC. For solutions that do not satisfy resource constraints, we will assign a sufficiently large fitness value. Since our goal is to minimize latency, IPSO will prefer to find solutions with less fitness.

We consider the position of each particle as a solution, and after evolution, IPSO obtains the optimal solution to the target. Next, the specific steps of IPSO are introduced.

Based on the formula proposed in [8], we divide the particles of each iteration into two types according to the probability, namely walkers and explorers:

$$\pi(X_j, i) = \frac{F(X_g)}{F(X_j)} * (\frac{i+N}{N}) \tag{16}$$

In the formula, $F(X_g)$ and $F(X_j)$ are the fitness of the global optimal particle and the jth particle respectively, and i and N represent the current iteration number and the maximum iteration number respectively. Particles with worse fitness values have lower $F(X_g)/F(X_j)$ values, which means they are more likely to play the role of walkers. On the other hand, particles with better fitness values have higher $F(X_g)/F(X_j)$ values, and are more suitable to play the role of the explorer. $\pi(X_j, i)$ represents the probability of playing the explorer, $1 - \pi(X_j, i)$ represents the probability of playing the walker. At this time, a random value a of $[0, 1]$ is generated and compared with $\pi(X_j, i)$. When $\pi(X_j, i) \leq a$, we select the particle as the explorer, otherwise we select the particle as the walker. For the two roles, we adopt different update strategies.

For the explorer, we use a linearly decreasing inertia weight, which balances the responsibilities of early exploration and late exploitation better than a fixed inertia weight.

$$w(i) = w_{max} - (w_{max} - w_{min})\frac{i}{N} \tag{17}$$

We set $w_{max} = 0.7$, $w_{min} = 0$. The velocity $V_j(i+1)$ of particle j is composed of three parts, the first is the inertial part $V_w(i)$, the second is the individual cognition part $V_c(i)$, and the third is the social cognition part $V_s(i)$.

$$V_j(i+1) = V_w(i) + V_c(i) + V_s(i) \tag{18}$$

where,

$$V_w(i) = w(i)V_j(i) \tag{19}$$

Algorithm 1: Improved Particle Swarm Optimization

```
Input: SFC requests $K$, iterations $N$
Output: global optimal fitness $F(X_g)$
Initialize a population of $n$ random solutions
for $i \leq N$ do
    Update the inertia weight $w(i)$
    for each particle $j$ do
        Calculate the fitness $F(X_j)$ of particle $j$ and update
        the particle's historical best position $pbest_j$
        Compare the fitness, update the global optimal position
        $gbest(i)$ and optimal fitness $F(X_g)$
    end for
    for each particle $j$ do
        According to Eq. (16), calculate the role-playing probability
        $\pi(X_j, i)$ and generate a random number $a \in [0, 1]$
        if $\pi(X_j, i) \leq a$ then
            The role of the particle is an explorer and
            update the position according to Eq. (18) and (22)
        else
            The role of the particle is a walker and update the
            position by randomly swapping the position value
            of the node to SFC $k \in K$
        end if
    end for
end for
```

$$V_c(i) = c_1 r_1 (pbest_j - X_j(i)) \tag{20}$$

$$V_s(i) = c_2 r_2 (gbest(i) - X_j(i)) \tag{21}$$

where, r_1 and r_2 are random numbers between [0, 1], $w(i)$ is the inertia weight, c_1 is the individual cognitive factor, and c_2 is the social cognitive factor. $gbest(i)$ represents the global optimal solution of the particle swarm, and $pbest_j$ represents the historical optimal solution of particle j. The location update formula is as follows:

$$X_j(i+1) = X_j(i) + V_j(i+1) \tag{22}$$

The position of particle j in the subsequent iteration is the sum of its position in the current iteration and its velocity in the next iteration. We adopted the node exchange strategy of the simulated annealing algorithm for walkers, which randomly exchanges nodes and expands the search range, thus allowing us to effectively escape local optimal solutions. The pseudo-code of the IPSO is presented in Algorithm 1.

3.2 TEG-Based SFC Routing

Once the Virtual Network Function (VNF) is deployed, we use solution X to carry out the routing of the Service Function Chain (SFC). Even under the

Algorithm 2: Delay Sensitive TEG-SFC Routing Algorithm

```
Input: TEG $G(N, E)$, Solution $X$, time slots $T$
Output: Routing $Path(X)$
for $t \leq T$ do
    while $\forall k \in K$ do
        Apply the Floyd-Warshall algorithm to get the shortest
        path for each stage of the $k$th SFC
        if the shortest path is feasible in the time slot $t$
            Choose the shortest path based on TEG
        else
            Calculate the storage delay $\delta^{s}_{k,n}$ and add the shortest
            path delay $\delta_0$ to get the actual delay $\delta_1$ of the path
            Calculate the delay $\delta_2$ of the shortest path based
            on the topology of the current time slot $t$
            if $\delta_1 \leq \delta_2$ then
                Choose the shortest path based on TEG
            else
                Choose the shortest path based on the current topology
            end if
        end if
    end while
end for
```

Time-Evolving Graph (TEG) condition, we can accurately predict the on-off situation of the link in the next time slot, but storage delays may still occur. To address this, we need an effective routing policy to guide the routing of the SFC.

Algorithm 2 outlines the specific steps for SFC routing. Initially, the inter-satellite link in the next time slot is assumed to be connected, and the shortest path is obtained using the Floyd-Warshall algorithm. If the links traversed by the shortest path are feasible in the current slot, then we choose the shortest path based on the TEG. If not, it means that we need to use a storage arc. The actual delay of the shortest path is the sum of the stored delay of entering the next time slot and the ideal path delay, and then it is compared with the shortest path delay based on the current slot topology to select a better path for SFC routing.

4 Simulation Results

In this section, we simulate on a Walk constellation network generated by STK software. In this network, 36 low-orbit satellites are distributed in 6 orbital planes, with an orbital inclination of 90° and an orbital altitude of 1000 km. The time slot length is set to 30 s, the time slot is set to 60. The specific simulation parameters are shown in Table 1.

We refer to the Dijkstra algorithm based on static topology as the Origin scheme, and the delay sensitive TEG-SFC routing algorithm as the DSTR

Table 1. Simulation parameters

Parameters	Numerical value
Data size	[100, 120] MB
Bandwidth resource capacity	[100, 200] Mbps
Computing resource capacity	[300, 500] MIPS
Computing Resource Requirements for VNFs	[50, 60] MIPS
Bandwidth Resource Requirements	[20, 25] Mbps
Number of SFCs	[1, 10]
Chain length of SFC	[4, 9]

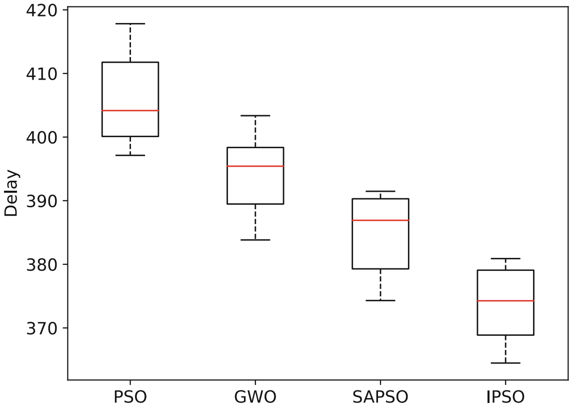

Fig. 3. Box plots of PSO, GWO, SAPSO, IPSO

scheme. We initially compare the performance of IPSO with the traditional heuristic algorithms PSO, GWO and the hybrid heuristic algorithm SAPSO (PSO algorithm with simulated annealing strategy). Figure 3 is a box diagram of four algorithms deploying eight SFCs with a chain length of 5 under the Origin scheme. The diagram reveals that the hybrid heuristic algorithm SAPSO has better performance than the traditional PSO and GWO, while the performance of IPSO is the most impressive.

The RANDOM algorithm in Figs. 4 and 5 is related to the random deployment of VNFs under the Origin scheme, while the PSO and IPSO algorithms also deploy VNFs under the Origin scheme. As seen in Fig. 4, the total service delay of the network increases with the number of SFCs, and THSDA is more effective in reducing delays than the other algorithms. We also compared the time delay spent by IPSO under different routing schemes, and the DSTR scheme was found to be superior to the Origin scheme. Figure 5 shows the comparison of link average loads of the four algorithms under different SFC numbers. It is evident that the average load of the link increases with the number of SFCs. Since THSDA takes into account different routing paths, it consumes fewer bandwidth resources than the three algorithms under the Origin scheme.

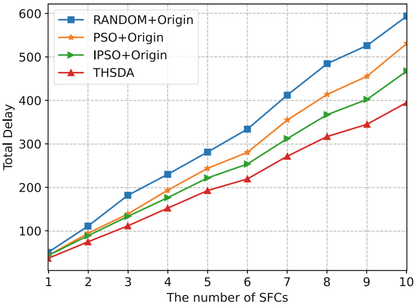

Fig. 4. The total delay with different number of SFCs

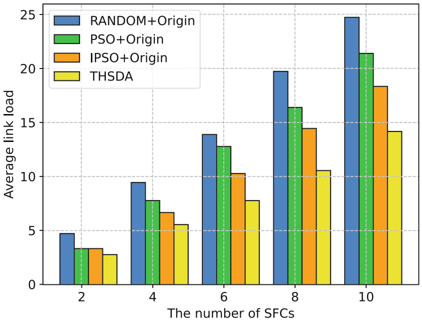

Fig. 5. The average link load with different number of SFCs

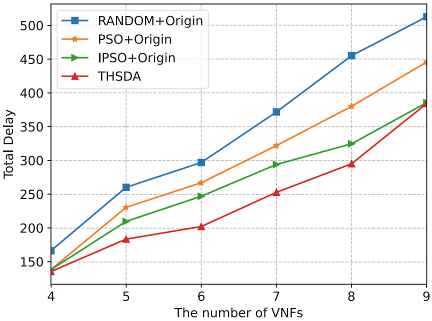

Fig. 6. Comparison of different algorithms with different number of VNFs

Figure 6 shows that the THSDA algorithm is capable of effectively reducing total delay under different lengths of SFC chains. It is consistently more successful than PSO and IPSO, especially as the chain length increases. When the chain length is short, the DSTR scheme is more successful in reducing delay. However, as the chain length increases, IPSO begins to reduce the delay more significantly, while the DSTR scheme becomes less effective. This is because the increase in SFC chain length leads to a change in the size of the storage delay, which affects the selection of the routing path. Additionally, the increase in chain length makes it more difficult to search for the optimal VNF deployment location, which is where the superiority of the IPSO algorithm is seen.

5 Conclusion

This paper investigates the challenge of deploying SFCs in SDSNs with the aim of minimizing service end-to-end delay. We present a heuristic approach to tackle this problem. Initially, the IPSO algorithm is employed to position the VNFs, and then the TEG-based routing strategy is used to pick the path, thus obtaining the optimal deployment plan of SFC. The simulation results demonstrate the superiority of the IPSO algorithm, and the THSDA algorithm can effectively reduce the total delay and decrease bandwidth usage.

References

1. Zhen, L., Bashir, A.K., Yu, K., Al-Otaibi, Y.D., Foh, C.H., Xiao, P.: Energy-efficient random access for LEO satellite-assisted 6g internet of remote things. IEEE Internet Things J. **8**(7), 5114–5128 (2020)
2. Ferrús, R., et al.: SDN/NFV-enabled satellite communications networks: opportunities, scenarios and challenges. Phys. Commun. **18**, 95–112 (2016)
3. Han, B., Gopalakrishnan, V., Ji, L., Lee, S.: Network function virtualization: challenges and opportunities for innovations. IEEE Commun. Mag. **53**(2), 90–97 (2015)
4. Liu, J., Shi, Y., Zhao, L., Cao, Y., Sun, W., Kato, N.: Joint placement of controllers and gateways in SDN-enabled 5G-satellite integrated network. IEEE J. Sel. Areas Commun. **36**(2), 221–232 (2018)
5. Fan, J., Guan, C., Zhao, Y., Qiao, C.: Availability-aware mapping of service function chains. In: IEEE INFOCOM 2017-IEEE Conference on Computer Communications, pp. 1–9. IEEE (2017)
6. Liu, Z., Li, Z., Liang, C., Liu, Z.: Routing and resource allocation for service function chain in service-oriented network. In: Gao, H., Wun, J., Yin, J., Shen, F., Shen, Y., Yu, J. (eds.) ChinaCom 2021. LNICST, vol. 433, pp. 465–480. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-99200-2_35
7. Yang, S., Li, F., Trajanovski, S., Yahyapour, R., Fu, X.: Recent advances of resource allocation in network function virtualization. IEEE Trans. Parallel Distrib. Syst. **32**(2), 295–314 (2020)
8. Xing, H., et al.: An integer encoding grey wolf optimizer for virtual network function placement. Appl. Soft Comput. **76**, 575–594 (2019)
9. Gao, X., Liu, R., Kaushik, A.: Service chaining placement based on satellite mission planning in ground station networks. IEEE Trans. Netw. Serv. Manag. **18**(3), 3049–3063 (2020)
10. Cai, Y., Wang, Y., Zhong, X., Li, W., Qiu, X., Guo, S.: An approach to deploy service function chains in satellite networks. In: NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium, pp. 1–7. IEEE (2018)
11. Wang, G., Zhou, S., Zhang, S., Niu, Z., Shen, X.: SFC-based service provisioning for reconfigurable space-air-ground integrated networks. IEEE J. Sel. Areas Commun. **38**(7), 1478–1489 (2020)
12. Yang, D., Wang, Z., Ding, H., Zhu, S., Meng, S., Duan, J.: Dynamic resource allocation of network function virtualization for space-air-ground integrated energy internet. In: 2021 International Conference on Power System Technology (POWERCON), pp. 1897–1903. IEEE (2021)
13. Jia, Z., et al.: Joint optimization of VNF deployment and routing in software defined satellite networks. In: 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall), pp. 1–5. IEEE (2018)
14. Hao, Y., Zhang, L., Zheng, L., Zhao, Z., Li, Z.: An efficient VNF deployment scheme for reliable service chain in software defined satellite networks. In: 2022 37th Youth Academic Annual Conference of Chinese Association of Automation (YAC), pp. 1363–1366. IEEE (2022)
15. Jia, Z., Sheng, M., Li, J., Zhou, D., Han, Z.: VNF-based service provision in software defined LEO satellite networks. IEEE Trans. Wireless Commun. **20**(9), 6139–6153 (2021)