Future Generation Computer Systems 119 (2021) 140–153

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Energy efficiency with service availability guarantee for Network Function Virtualization

Liuyang Mai [a], Yi Ding [a], Xiaoning Zhang [a,*], Lang Fan [a], Shui Yu [b], Zhichao Xu [a]

[a] School of Information and Communication Engineering, University of Electronic Science and Technology of China, China
[b] School of Software, University of Technology Sydney, Australia

ARTICLE INFO

Article history:
Received 12 September 2020
Received in revised form 5 January 2021
Accepted 5 February 2021
Available online 10 February 2021

Keywords:
Network Function Virtualization
Service function chain
Reliability
Protection
Lagrangian relaxation

ABSTRACT

Following the trend of Network Function Virtualization (NFV), dedicated hardware middleboxes are replaced with innovative and flexible software middleboxes also known as Virtual Network Functions (VNFs). An ordered sequence of VNFs composing a logical service is called Service Function Chain (SFC). VNFs are generally run on commodity servers. In this way, the capital and operational expenditures of buying and maintaining dedicated hardware for telecom operators can be greatly reduced. One of the key issues in NFV is the optimal VNF placement and service chaining to achieve energy efficiency. However, the current NFV energy saving approaches seem to consider energy minimization as the only objective to be optimized. Little or no attention is given to other important aspects, e.g., service availability, which is paramountly important to fulfill Service Level Agreement (SLA) for telecom operators. This paper investigates the energy efficiency optimization with service availability guarantee in NFV-enabled networks. We firstly propose a novel green orchestration NFV architecture. Then, an energy-efficient VNF placement framework guaranteeing service availability is presented under the proposed architecture, and evaluated by extensive simulations. Open research issues and technical challenges in this emerging area are also presented.

1. Introduction

Network Function Virtualization (NFV) [1] is a promising concept proposed by the European Telecommunication Standards Institute (ETSI) to simplify the deployment and management of networking services using virtualization technologies. Compared with traditional hardware appliances, NFV enables service deployment on virtual machines to enhance the flexibility and scalability. Given such promising benefits, NFV has already gained a considerable momentum, with many prototype development and standardization efforts ongoing [2].

The basic idea behind NFV is decoupling network functions (such as firewall, load balancer, and proxy, etc.) from proprietary hardware appliances so they can run in software. It is attracting more attention as a promising architecture. Under such a structure, the network function needs to be deployed flexibly and dynamically, without being restricted by the physical network. By using commodity hardware (e.g., x86 based systems), Virtual Network Functions (VNFs) are no longer dependent on dedicated hardware [3]. In NFV-enabled networks, traffic is often required to pass through and processed by an ordered sequence of VNFs called Service Function Chain (SFC). For instance, traffic may need to pass through an intrusion detection system, then a load balancer and finally a firewall.

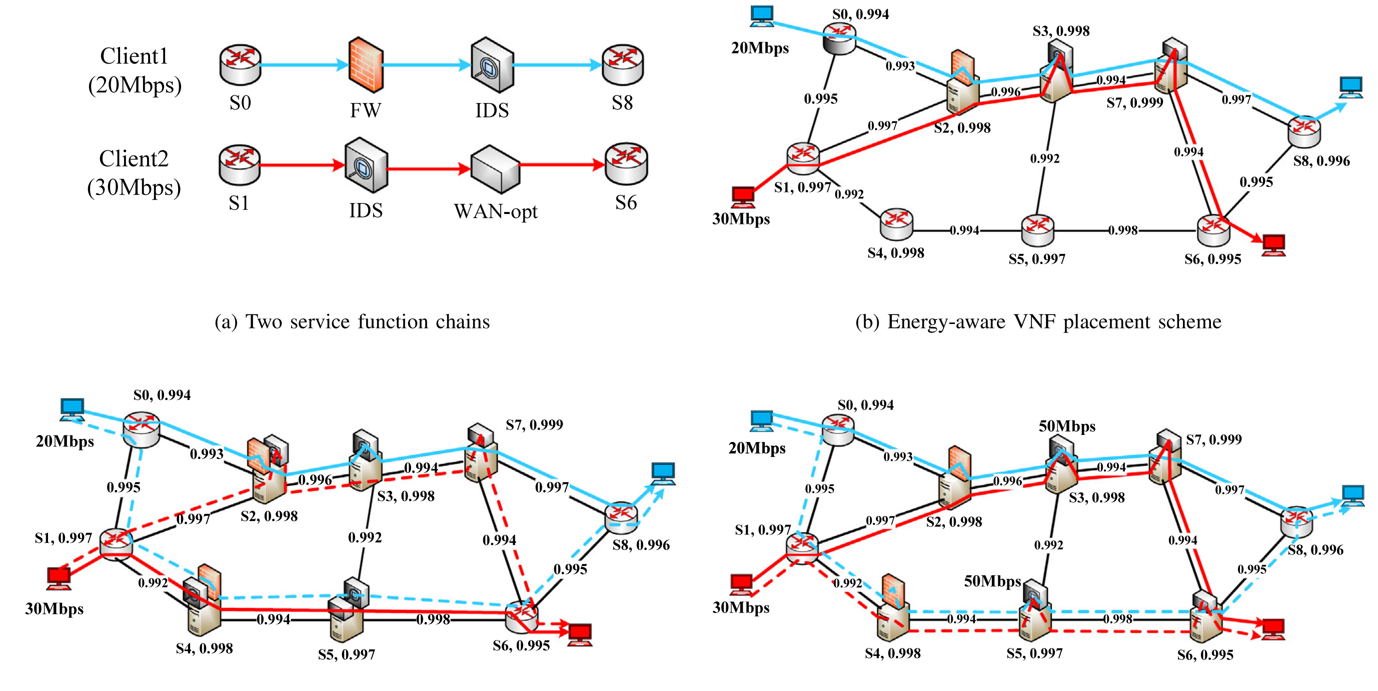
One of key issues in NFV research is energy efficient design for VNF placement and service chaining. The ETSI emphasizes the importance of energy efficiency and environmental sustainability by proposing a standard called Green Abstraction Layer (GAL) [4] enabling operators to manage the energy consumption of telecom networks. ETSI also indicates that the energy consumption of NFV is dominated by the degree of resource sharing. Existing studies indicate that NFV can reduce up to 50% energy consumption compared with traditional appliance based telecom networks [5].

As above, energy efficient design for NFV-enabled networks has not been seriously considered, especially with service availability [6], which is paramountly important to fulfill Service Level Agreement (SLA) for telecom operators. In fact, NFV aggravates the end-to-end service availability problem since VNFs are deployed over virtualized and shared platforms on commodity servers. Even if the availability of each VNF and physical link is high, the availability of a SFC may be unacceptable. Therefore, it is unwise to consider energy efficient VNF placement without end-to-end service availability guarantee.

We give an example in Fig. 1 to illustrate our motivation. In order to save energy, servers and links do not consume energy if they are just used for backup. We assume that a non-idle physical link consumes 1 energy unit while server node consumes

* Corresponding author.
E-mail address: xnzhang@uestc.edu.cn (X. Zhang).

https://doi.org/10.1016/j.future.2021.02.002
0167-739X/



(a) Two service function chains　(b) Energy-aware VNF placement scheme

(c) Availability-aware VNF placement scheme　(d) Improved VNF placement scheme

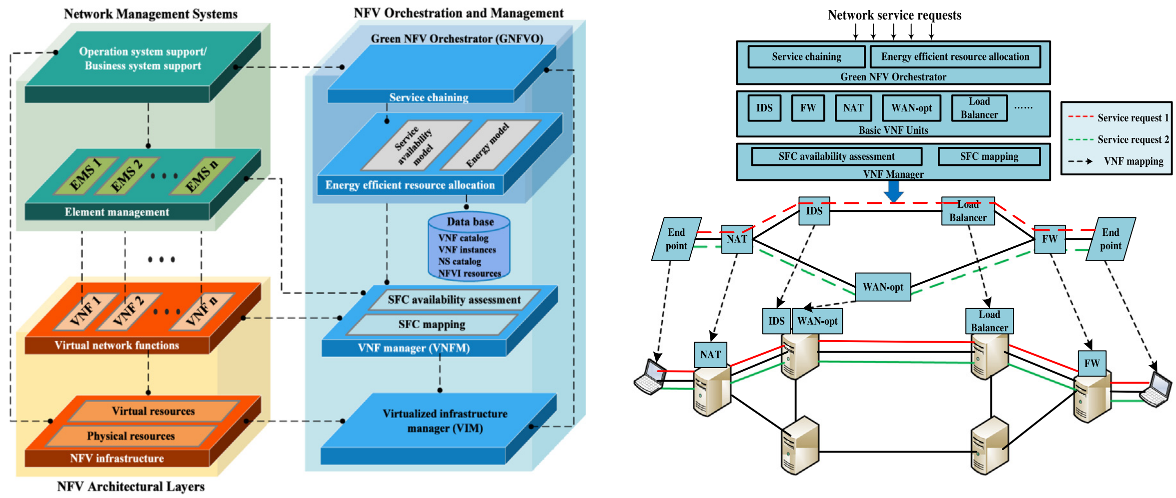
Fig. 1. Example of VNF placement schemes.

2 energy units, respectively. The availability requirement of SFC is 0.999. Fig. 1(a) depicts two SFCs: Client 1 and Client 2. In Fig. 1(b), the energy-aware VNF placement scheme tries to aggregate the two SFCs onto the same path to minimize energy consumption. Thus, the availability of Client1 and Client2 is 0.97623 and 0.97819, respectively; and the total energy consumption is 12 energy units. In Fig. 1(c), the availability-aware VNF placement scheme leverages the off-site [7] backup method to improve and thus guarantee the availability requirement. In off-site backup scheme, we let some other server nodes and corresponding physical paths be backup for SFCs and pre-allocate equal amount of computing resources and bandwidth resources on them. Once network breakdown occurs, all affected traffic will be transferred to those backup VNFs and routing paths, and continue to work. As shown, the FW of SFC1 on S2 is protected by the pre-allocated backup FW on S4; the IDS of SFC1 on S3 is protected by backup IDS on S5; the IDS of SFC2 on S4 is protected by backup IDS on S2; the WAN-optimizer of SFC2 on S5 is protected by the backup server S7. The backup paths for the two SFCs are both drawn with dashed line in Fig. 1. (c). Thus, the availability of Client1 and Client2 with backup path is 0.99939 and 0.99953, respectively; but the total energy consumption increases to 17 energy units. In Fig. 1 (d), the improved VNF placement scheme steers the two SFCs to share the IDS on S3. In addition, the IDS on S3 is protected by the backup IDS S5 with 50 Mbps computing resources, and the FW on S5 and WAN-optimizer on S7 are protected by backup FW and backup WAN-optimizer on S4, S6, respectively. Thus, the availability of Client1 and Client2 with backup path is 0.99939 and 0.99948, respectively; but the total energy consumption is only 12 energy units. We can see that the improved VNF placement scheme consumes less energy consumption while guaranteeing service availability.

In this paper, we investigate how to minimize energy consumption with service availability guarantee for NFV. Specifically, we study the problem of energy-efficient VNF placement with availability. We first propose a novel green orchestration NFV architecture, which automatically manages virtual resources for energy efficiency. Then we illustrate the energy model and service availability model for NFV-enabled networks, and we formulate the research problem as an Integer Linear Programming (ILP) model. Since the studied problem is NP-hard, we present a heuristic algorithm based on Lagrange relaxation technique. The technical implementation of our proposed architecture is dependent on this algorithm. Our algorithm has been evaluated on the typical carrier network. Extensive simulations show that our algorithm can save up to 10.57% of energy consumption in average compared to previous benchmark algorithms. In addition, our algorithm provides the highest availability for services while guaranteeing the availability requirements of services. The main novel contribution of this paper are summarized as follows.

- For the problem of energy-efficient with service availability guarantee for NFV, we propose a novel green orchestration NFV architecture, which can solve the problem of energy-efficient VNF placement with availability.
- We formulate the studied problem as an ILP model and prove the NP-hardness of problem. Since the studied problem is NP-hard, we present an efficient heuristic algorithm based on Lagrange relaxation technique.
- We evaluate the performance of the proposed algorithms with extensive simulations and the simulation results show that our algorithm can save up to 10.57% of energy consumption compared to previous benchmark algorithms while guaranteeing the availability requirements of services.

The remainder of this paper is organized as follows. In Section 2, we briefly discuss the related work. After that, we propose a novel green orchestration NFV architecture in Section 3. In Section 4, we present the system model and prove the studied problem is NP-hard. Then, in Section 5, we design an efficient algorithm as the core technology of our proposed architecture to solve the studied problem. We evaluate the performance of our proposed algorithm by using typical carrier network topology in Section 6. At last, we give conclusion and extend with some open researches in Section 7.



(a) Three main entities of Green Orchestration NFV Architecture

(b) A use case under the green orchestration NFV Architecture

Fig. 2. Illustration of green NFV orchestration architecture.

2. Related work

Recently, VNF placement problem has been widely covered in some literatures. In practice, the highest cost in telecom networks is not related to personnel or construction, but to energy consumption of network equipments [8]. Hence, there have been some efforts on the energy-efficient VNF placement problem. Zakeri et al. [9] considered both the access and core parts of the network and defined the network cost as the consumed energy and the number of utilized network servers. Then they presented a novel heuristic algorithm for embedding and scheduling of VNFs by proposing a novel admission control (AC) algorithm, and used alternative search method (ASM) to solve it. Soualah et al. [10] proposed the Monte Carlo Tree Search (MCTS) method to achieve energy efficiency in NFV-enabled cloud platform. The method uses physical resource consolidation and shares VNFs between multiple tenants. Pham et al. [11] proposed a novel approach named as joint sampling-based Markov approximation and matching-theoretic approach to find an efficient solution for purpose of energy and traffic-aware cost minimization. Ahvar et al. [12] proposed the cost-efficient centrality-based VNF placement and chaining algorithm. The objective of the algorithm is to find the optimal number of VNFs along with their locations, in such a manner that the network service provider cost is minimized. Tipantuna et al. [13] proposed a long-term sustainable Demand–Response (DR) architecture for the efficient management of available energy consumption for Internet of Things (IoT) infrastructures. The proposal used Network Functions Virtualization (NFV) and Software Defined Networking (SDN) technologies as enablers and promotes the primary use of energy from renewable sources. However, all the above works do not consider the factor of service availability, which is non-negotiable for telecom industry since five nines (99.999%) end-to-end availability is the benchmark.

The availability-aware VNF placement problem has been studied in some works. Zhong et al. [14] proposed a Cost-aware and Reliability-guaranteed SFC Orchestration (CRSO) scheme. CRSO firstly orchestrates SFCs in inter-DC network with less cost based on Hidden Markov Model (HMM). Then it backups VNFs to satisfy the reliability requirements. Khezri et al. [15] mathematically modeled the problem of jointly minimizing the placement cost and maximizing the number of admitted services based on the reliability requirement. They presented a solution based on DQN for reliability-aware NFV placement and employed Deep Reinforcement Learning (Deep-RL) to improve the performance of the network operator. In [16], redundancy mechanism is proposed as an effective solution to improve service reliability. The method can select qualified backups and map them to high reliability nodes for accomplishing deployment. In [17], the authors proposed a genetic algorithm, the algorithm can optimize the placement cost while providing queue-aware dynamic placement scheme for VNFs. Yang et al. [18] investigated how to quantitatively model the traversing delay of a flow in both totally ordered and partially ordered SFCs. They calculated the VNF placement availability mathematically for both unprotected and protected SFCs and studied the delay-sensitive Virtual Network Function (VNF) placement and routing problem with and without resiliency concerns. Then they evaluated the proposed algorithms in terms of acceptance ratio, average number of used nodes and total running time via extensive simulations. Torkamandi et al. [19] proposed a new Availability-aware Clustered SFC Embedding (AC-SFC) algorithm which not only satisfies the availability requirement of SFC but also reduces the footprints of backup resource by using the Share Protection Cluster (SPC). These studies focus on service's availability with performance issues, and ignore the influence of energy consumption in NFV. As mentioned above, energy consumption in NFV cannot be ignored.

From the above, the availability problem and energy consumption problem of NFV have been studied respectively in some literatures. Nevertheless, there are few literatures studying on the problem of optimization of energy consumption with guarantee service availability. In fact, these two indicators are theoretically contradictory. In order to increase the availability of the links, the energy consumption is bound to increase meanwhile in order to control the energy consumption, the availability is bound to be sacrificed. The existing researches mostly use better VNF placement algorithms to unilaterally reduce energy consumption or increase service availability. In this paper, we first study how to minimize energy consumption while guaranteeing service availability in NFV-enabled networks. It is a step towards the practical use of NFV energy efficient approaches for telecom operators.

3. Green Orchestration NFV architecture

In this section, we present a novel green orchestration NFV architecture. This architecture performs energy efficient VNF placement while guaranteeing network performance, such as end-to-end availability, delay, traffic throughput, etc.

Fig. 2(a) illustrates the green orchestration NFV architecture. According to the ETSI MANO (Network Functions Virtualization Management and Orchestration) framework [20], the functional blocks in the green orchestration NFV architecture can be grouped into three main entities:

1. Operation System Support/Business System Support (OSS/BSS) and Element management compose the network management systems, which integrate NFV technical architecture with existing OSS/BSS;
2. Virtual network functions and NFV infrastructure compose the NFV architecture layer, which provides VNF deployment environment;
3. Virtualized Infrastructure Manager (VIM), VNF Manager (VNFM) and Green NFV Orchestrator (GNFVO) compose the NFV orchestration and management, which manages and orchestrates VNFs and other software components for the purpose of energy efficiency.

Obviously, the GNFVO is the key function block in the green orchestration NFV architecture. It binds VNFM and VIM together to create an end-to-end, resource-coordinated, and energy-efficient service chains. The GNFVO consists of two modules: service chaining and energy efficient resource allocation. The service chaining module provides the capability to create a SFC by composing multiple VNFs. In this way, safe, fast and stable network services can be provided to clients while meeting diverse requirements. Based on the energy model and service availability model of SFC, the energy efficient resource allocation module provides a energy efficient VNF deployment and traffic steering solution for network service requests while guaranteeing network performance, such as end-to-end availability. Finally, the GNFVO implements the orchestration solution on NFV-enabled networks via the VNFM and the VIM.

Fig. 2(b) illustrates a use case under the green orchestration NFV architecture. In this figure, two service requests are mapped to the NFV-enabled network for the purpose of energy efficiency. In the mapping procedure, a service request are first served by the GNFVO, which creates a service chain by connecting VNF instances in the data base, and allocates virtual resource for the service to minimize the energy consumption. Then, VNF manager implements the mapping solution from the GNFVO by allocating VNFs and virtual resources of NFV infrastructure. In this case, the two service requests are aggregated on the same path in the physical network, and the IDS instance and WAN-opt instance are both deployed on the same server. By this way, the total energy consumption of the NFV-enabled network can be minimized.

4. Energy efficient VNF placement framework with service availability guarantee

In this section, we propose the energy efficient VNF placement framework with service availability guarantee. In addition, we will give an Integer Linear Programming (ILP) model for the problem. All the notations used in this section are summarized in Table 1.

4.1. Problem statement

We consider a NFV-enabled network $G = (V, L)$, which consists of node (NFV-enabled server) set V and physical link set L. Without loss of generality, we assume all the servers are homogeneous, and each of them is equipped with C_v computing capacity. Each link is equipped with C_l capacity, which means that each link can let C_l traffic volume pass through at most. Given the end-to-end service availability requirement and bandwidth of the set of network service requests R. Our objective is to minimize energy consumption while guaranteeing service availability in NFV-enabled networks.

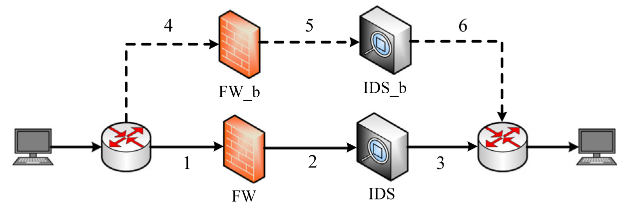


Fig. 3. Illustration of off-site backup for SFC.

4.2. Energy model

In NFV-enabled networks, servers hosting VNFs and physical links carrying traffic consume electrical energy. To evaluate the energy consumption of NFV-enabled networks, we present the energy model for servers and physical links.

The energy consumption of a server consists of two main parts based on the hybrid energy model [21]: (1) the energy consumption for keeping itself on denoted as p_b^v, and (2) the energy consumption produced by processing network service requests, which is positively related to computing resource utilization. We denote the peak-load energy consumption of server v as p_h^v and the consumption of processing requests as $load_v$. Then the energy consumption of a server p_v can be calculated with:

$$p_v = p_b^v + (p_h^v - p_b^v) \cdot \frac{load_v}{C_v} \quad \forall v \in V \tag{1}$$

For the energy consumption of physical links, the energy consumption of a physical link depends on its on/off state and bandwidth utilization [22]. Let p_b^l denote the startup energy consumption of link l and p_h^l denote its peak-load energy consumption. The consumption of communication resources is denoted as $load_l$. Then the energy consumption of a link can be calculated with:

$$p_l = p_b^l + (p_h^l - p_b^l) \cdot \frac{load_l}{C_l} \quad \forall l \in L \tag{2}$$

Therefore, the total energy consumption of a NFV-enabled network is the sum of the energy consumption of all servers and physical links.

4.3. Service availability model

The availability of SFC is defined in the SLA along with the service reliability in NFV-enabled networks. To meet the service availability requirement, Fan et al. [7] proposed an off-site backup scheme for VNFs to improve the availability of SFC. In the scheme, operators compute the minimum number of off-site backup VNFs needed to guarantee a certain degree of availability of a SFC. Afterwards, the location is decided to reserve redundant computing resources for backup VNFs, and consider sharing among multiple backup VNFs. In our study, we assume that a VNF can serve multiple SFCs if it has equal or more computing resources than the sum capacity demands of these SFCs.

A SFC consists of a series of ordered virtualized network functions and virtual links connecting them. For example, as shown in Fig. 3, there is a SFC consisting of two VNFs: FW and IDS. Apparently, the availability of an unprotected path is the product of the availability of all VNFs and physical links connecting them. To evaluate the availability for the given SFCs, we formulate the availability computing model. In our study, we assume that the failure of each VNF and link is independent. To meet the availability requirement, two backup VNFs (i.e., *FW_b* and *IDS_b*) are placed parallelly to backup for FW, IDS, respectively. Therefore,

Table 1
Notations used in the paper.

Notaion	Description
$G(V, L)$	The network topology, where V denotes the set of nodes and L denotes the set of links
C_v	The computing capacity of server
C_l	The capacity of link
F	The set of VNFs, with function $f \in F$
v, u	The notes in the network, where $v, u \in V$
l	The physical link in the network, where $l \in L$
R	The set of SFC requests, with SFC request $r \in R$
b_r	The bandwidth of r
F_r	The set of ordered VNFs for r, where $F_r \subseteq F$
$z_{r,f}^{v}$	A binary variable indicating whether r goes through function f on node v, where $f \in F_r$
w_r^{vfug}	A binary variable indicating whether r traverses from function f on node v to function g on node u, where $f, g \in F_r$ and $v, u \in V$
x_r^{fg}	A binary parameter indicating whether function g is the successor of function f in F_r, where $f, g \in F_r$
y_r^{uvl}	A binary variable indicating whether r traverses through link l from node v to node u
$load_v$	The consumption of processing requests in server v
$load_l$	The consumption of processing requests in link l
p_v	The energy consumption of server v
p_l	The energy consumption of link l
p_b^v	The basic energy consumption for starting up server v and maintaining it
p_h^v	The peak-load energy consumption of server v
p_b^l	The startup energy consumption of link l
p_h^l	The peak-load energy consumption of link l
p_total	The energy consumption of all servers and links in the network
$d_r(F)$	The services availability of the VNF set that request r contains
$d_r(L)$	The services availability of the link that request r passes
d_f	The services availability of the function f
d_l	The services availability of the link l
δ_{rel}	The availability requirement of SFC request

the availability of this SFC denoted as A_{SFC} can be calculated with:

$$A_{SFC} = 1 - (1 - A_1 \cdot A_{FW} \cdot A_2 \cdot A_{IDS} \cdot A_3) \times (1 - A_4 \cdot A_{FW\_b} \cdot A_5 \cdot A_{IDS\_b} \cdot A_6), \tag{3}$$

where A_{FW}, A_{IDS}, A_{FW_b}, A_{IDS_b} denotes the availability of FW, IDS, backup VNF *FW_b* and *IDS_b*, respectively. $A\_i, i = 1, 2...6$ denotes the availability of physical link i. However, in reality, only one backup path for SFC cannot meet the harsh availability requirement. Thus, we propose a Lagrangian-based algorithm to find a set of backup paths while minimizing energy consumption.

4.4. Integer linear programming formulation

As energy consumption and availability is two important metric of network, the optimization goal of this problem is to minimize the total energy consumption of the NFV telecommunications network while guaranteeing the requirement of availability. In this section, we will establish an Integer Linear Programming (ILP) model for Availability guarantee Energy Efficient VNF Placement (AEEVP).

Let F denote the set of VNFs, and each node in the network is deployed a instance of function $f (f \in F)$. In addition, r denotes the SFC request, and F_r denotes the ordered network function sequence of r.

We first define a decision binary variable z_r^{vf} to denote that if SFC request r is processed by function f on node $v (f \in F_r)$:

$$z_r^{vf} = \begin{cases} 1 & \text{if } r \text{ is processed by function } f \text{ on node } v \\ 0 & \text{otherwise} \end{cases}$$

Next, we define another decision binary variable w_r^{vfug} to denote if SFC request r traverses from function f on node v to function g on node $u (f, g \in F_r$ and $v \in V_v, u \in V_g)$:

$$w_r^{vfug} = \begin{cases} 1 & \text{if } r \text{ traverses from } f \text{ on node } v \text{ to } g \text{ on } u \\ 0 & \text{otherwise} \end{cases}$$

Then, the third binary decision variable x_r^{fg} is defined to indicate if function g is the successor of function f in $F_r (f, g \in F_r)$:

$$x_r^{fg} = \begin{cases} 1 & \text{if function } g \text{ is the successor of } f \text{ in } F_r \\ 0 & \text{otherwise} \end{cases}$$

Additionally, we define the fourth binary decision variable y_r^{uvl} denoting if SFC request r traverses through link l from node u to node v:

$$y_r^{uvl} = \begin{cases} 1 & \text{if } r \text{ traverses through link } l \text{ from node } u \text{ to } v \\ 0 & \text{otherwise} \end{cases}$$

Finally, according to the service availability model which is presented above, we define $d_r(F)$ to indicate the VNF services availability of the request r and $d_r(L)$ for the services availability of the link passed. We define d_f and d_l to denote the availability of the function f and the link l.

$$d_r(F) = \prod_{f \in F_r, x_r^{vf} \neq 0} x_r^{vf} d_f$$

$$d_r(L) = \prod_{u,v \in V, l \in L, y_r^{uvl} \neq 0, w_r^{vfug} \neq 0} w_r^{vfug} y_r^{uvl} d_l$$

At first, for illustration, two virtual VNF nodes are added at the head and tail of each SFC request, denoted as D_0 and D_1 respectively. To meet the constraints of traffic conservation, SFC should start from D_0 and end with D_1. Accordingly, the constraint can be expressed as:

$$\sum_{u,v \in V, f, g \in F_r} w_r^{vfug} - \sum_{u,v \in V, f, g \in F_r} w_r^{ugvf} = \begin{cases} 1 & \text{if } f = D_0 \\ -1 & \text{if } f = D_1 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$
$$\forall r \in R, \forall f, g \in F_r, \forall u, v \in V$$

We assume that SFC request r passes through function f (where $f \in F_r$) only once:

$$\sum_{v \in V} z_r^{vf} = 1 \quad \forall f \in F_r, \forall r \in R \tag{5}$$

As discussed in Section 2, SFC request r must traverse the VNFs in a required order described by x_r^{fg}. Thus, w^{vfug} should be restricted as:

$$\sum_{u,v\in V} w^{vfug} = x_r^{fg} \quad \forall f,g \in F_r, \forall r \in R \tag{6}$$

In addition, we also have to guarantee the capacity constraint of nodes and physical links:

$$\sum_{r\in R}\sum_{f\in F_r} z_r^{vf} b_r \leqslant C_v \quad \forall v \in V \tag{7}$$

$$\sum_{r\in R} w^{vfug} y_r^{uvl} b_r \leqslant C_l \quad \forall l \in L, \forall u,v \in V \tag{8}$$

Considering the availability requirement of SFC request:

$$d_r(F)d_r(L) \geqslant \delta_{rel} \quad \forall r \in R \tag{9}$$

Based on the above constraints, the following ILP model is established for the AEEVP problem:

$$Minimize: p_{total} = \sum_{v\in V} p_v + \sum_{l\in L} p_l \tag{10}$$

Subject to: (4)-(9)

Due to the complexity of the problem, we introduce a heuristic algorithm in Section 5.

4.5. Problem analysis

This problem can be reduced to Virtual Network Embedding problem [12] by relaxing the constraint of services availability. The Virtual Network Embedding problem has been proved as an NP-hard problem. Thus, the AEEVP problem is a NP-hard problem.

To prove NP-hardness of our problem, we first illustrate VNE problem. Then the Availability guarantee Energy Efficient VNF Placement problem can be redefined and reduced from the VNE problem. In VNE problem, the substrate network topology can be modeled as an undirected graph $G_P(U_P, E_P)$, where U_P denotes the set of substrate nodes and E_P denotes the set of substrate links. Each node $u_i \in U_P$ is assigned with a capacity $C_P(u_i)$ and each link $(u_i, u_j) \in E_P$,(where $u_i, u_j \in U_P$)has a bandwidth capacity $b_P(u_i, u_j)$. Given another undirected graph $G_V(W_V, E_V)$ representing virtual network, where W_V denotes the set of virtual nodes, and E_V denotes the set of virtual links. Each virtual node $w_k \in W_V$ is assigned with a capacity $C_V(w_k)$ and each link (w_k, w_l) (where $w_k, w_l \in E_V$) has a bandwidth capacity $b_V(w_k, w_l)$.

The VNE problem is actually to determine whether a set of valid mappings from the edge set E_V to the edge set E_P can be found. For each pair $w_k, w_l \in E_V$ to $u_i, u_j \in U_P$, two constraints need to be met:

1. $C_P(u_i) \geq C_V(w_k)$, and $C_P(u_j) \geq C_V(w_j)$;
2. $b_P(u_i, u_j) \geq b_V(w_k, w_l)$.

Theorem 1. *The AEEVP problem is NP-hard.*

Proof. Without considering the constraint of reliability and energy consumption, we simplify our AEEVP problem as an easier virtual network function placement (VNFP) problem in polynomial time. First, the physical network is represented by an undirected graph $G(N, L)$, where N is the set of servers, and L represents the set of physical links. Each server node $u \in N$ and physical link $(u, v) \in L$ is assigned a capacity attribute $C^N(u)$ and $C^L(u, v)$, respectively. Next, construct a virtual network diagram for each SFC request. The undirected graph $G^V(N^V, L^V)$ is used to represent this virtual network graph, where N^V represents a collection of virtual nodes, and L^V represents a collection of virtual links. In addition, a virtual node refers to a VNF instance, and a virtual link refers to a path connecting two VNF instances in an SFC. Each VNF instance has a processing capacity requirement C_f, which is used to indicate the computing resources required to create and enable a VNF instance $f \in F_v$, where F_v represents the set of VNFs that need to be deployed. Each virtual link has a link bandwidth resource requirement $d_{a,b}(u, v)$, which represents the amount of bandwidth resources that the virtual link (a, b) needs to consume on the physical link (u, v), where $(a, b) \in L^V$, $(u, v) \in L$.

Since multiple virtual nodes can be mapped to a single physical node of the network, at physical node u, the sum of processing power requirements of all virtual nodes mapped to u must be no more than the maximum capacity of quantity u. Similarly, since multiple virtual links can be mapped to a single physical link, the sum of the bandwidth requirements of the virtual links mapped to the physical link must be no more than the maximum bandwidth capacity of the physical link. For each pair of virtual to physical mapping, all $a \in N^V$ mapped to $U \in N$, all $B \in N^V$ mapped to $v \in N$, and all links $(a, b) \in L^V$ mapped to $(u, v) \in L$ must satisfy the following constraints condition:

1. $\sum_{f\in F_V} C_f(u) \leq C^N(u)$, $\sum_{f\in F_V} C_f(v) \leq C^N(v)$, $\forall u, v \in N$, where $\sum_{f\in F_V} C_f(u)$ and $\sum_{f\in F_V} C_f(v)$ respectively represent the sum of the capacity demands of all virtual nodes mapped to physical nodes u and v.
2. $\sum d_{a,b}(u, v) \leq C^L(u, v)$, $\forall (a, b) \in L^V$, where $d_{a,b}(u, v)$ represents the sum of bandwidth resource requirements of all virtual links mapped to physical links (u, v).

Definition 1. function f: $\delta_1 \to \delta_2$ is called a *mapping reduction* from X to Y if:

(a) for any $\beta \in \delta_1$, $\beta \in X$ if $f(\beta) \in Y$.
(b) f represents a polynomial function.

Therefore, the mapping reduction from problem X to problem Y means that a computer can convert any instance of X into an instance of Y, and then the solution of Y is the solution of X. Finally, the following mapping relationship can be established between the variables of the VNE problem and the EEVP problem:

$$\left\{\begin{array}{l} C_P(u_i) \to C^N(u) \\ C_P(u_j) \to C^N(v) \\ C_V(w_k) \to \sum_{f\in F_v} C_f(u) \\ C_V(w_l) \to \sum_{f\in F_v} C_f(v) \\ b_P(u_i, u_j) \to C^L(u, v) \\ b_V(w_k, w_l) \to \sum d_{a,b}(u, v) \end{array}\right\} \tag{11}$$

Through the above Definition 1 and Eq. (11), the problem of VNE, which has been proved to be NP-hard, can be mapped and reduced to the simplified VNFP problem. Since the AEEVP optimization problem is a VNFP problem considering energy consumption and service availability, it is obvious that the AEEVP problem is more difficult than the VNFP problem. Therefore, the AEEVP problem studied in this paper is also NP-hard.

5. Lagrangian Relaxation based heuristic algorithm

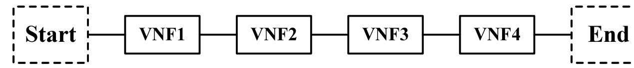
In this section, we will first introduce the VNF off-site backup mechanism adopted in this study, then introduce the theory of Lagrangian relaxation technology, and finally design a heuristic algorithm based on Lagrangian relaxation technology to guarantee end-to-end service availability while minimizing energy consumption of the network.

5.1. Redundant backup protection mechanism

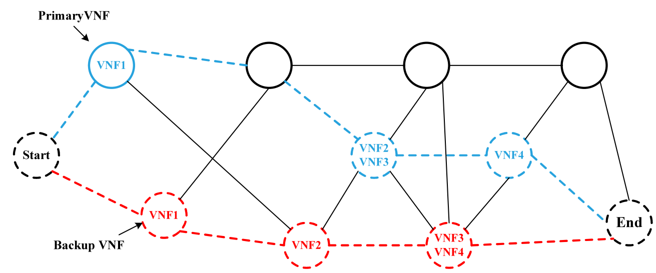
An important aspect of network operators' concern is to ensure continuity of service in the event of a network failure. To achieve this goal, network resilience must be taken into account during the design phase, which motivates us to adopt the idea of deploying redundant VNF instances. Redundant VNF instances typically remain in standby mode, but they will be turned to active mode when network failures (e.g., failure of the primary VNF node or failure of physical link) happen. When it is activated, the traffic affected by the failure is immediately switched and led to the backup network device. The redundancy scheme depends on the type of fault. According to the Ref. [23], in this section, three protection schemes are introduced, which are further divided into the following two categories:

1. Local Redundant Backup: In order to meet the expectations of customer service requirements for latency, critical VNFs supporting critical services need to quickly switch to backup VNF to ensure service availability, thus it is necessary to instantiate the backup VNF locally (i.e., centralized redundancy). As result, critical VNFs may require 1+1 redundancy, while less critical functions can tolerate 1:1 redundancy. The main benefit of centralized redundancy is reduced switching time, which speeds up the service recovery process and reduces the amount of internal state information that needs to be transferred from the primary VNF to the backup VNF. Since the primary VNF and the backup VNF are deployed at the same physical location in this scheme, the resilience for node failure is not guaranteed.
2. Off-site Redundancy Backup: The off-site redundancy backup mechanism places the redundant VNF on the selected remote location or on the NFVI node in network operator service area and puts it in (hot or cold) standby mode. The goal is to instantiate these redundant VNFs when failure happens on one of the many NFVI-Points-of-Presence (NFVI-PoP). In this way, since the backup VNF is not in the same physical location as the primary VNF, this scheme can guarantee the resilience for both link and node failure. Therefore, based on the criticality of the service and the pertinence of the resilience guarantee, operators can choose between local or off-site redundancy backup schemes according to their needs.

This paper uses end-to-end protection (E2E-P) for the entire SFC request based on the off-site redundancy backup scheme. In order to provide SFC with the ability to withstand single-link and single-node failure, the backup path for SFC, which is composed of VNFs deployed at different physical locations, is pre-located in physical network to protect primary VNFs and routing path. In this way, end-to-end service availability is promoted by switching traffic when failures happen. In this protection scheme, the paths linking primary VNFs and backup VNFs are not allowed to intersect. An example of such a protection scheme is shown in Fig. 4(b), where the SFC shown in Fig. 4(a) consisting of four VNFs needs to be placed in physical network. This protection scheme is an off-site redundancy backup strategy because all backup VNFs are instantiated in different locations from the primary VNF. In this case, both 1+1 and 1:1 redundancy strategies are possible, depending on the service delay requirements and the operator's design goals in terms of resource utilization.



(a) SFC to be deployed



(b) End-to-end protection

Fig. 4. Off-site End-to-end protection scheme.

5.2. Lagrangian Relaxation technique

For the first time, Held and Karp proposed the Lagrangian relaxation technique for the traveling salesman problem in the literature [24,25]. This technique is often used to calculate the lower bounds and this paper leverages it to derive effective solution for our studied problem.

In the problem studied in this paper, we leverages the Lagrangian relaxation technique to find the optimal routing path $p, p \in P(s,t)$ that minimizes the energy consumption increment $C(p)$ and guarantees the availability requirement δ_{rel} for each SFC request. Then, the cost optimization function can be expressed as:

$$\min\{C(p), p \in P(s,t) \quad d(p) \geq \delta_{rel}\}, \tag{12}$$

where $P(s,t)$ is a set denoting all the available routing path between source node s and destination node t, $d(p)$ is the end-to-end service availability of routing path p calculated based on previous availability models.

This section leverages the technology to solve the energy efficient VNF placement with availability guarantee (AEEVP) problem. Thus we designed a heuristic algorithm in this section based on Lagrangian relaxation technique, which aggregates energy costs and revises the cost function by minimizing $C_\lambda := C - \lambda \cdot d$. For a given (fixed) λ, the path with the least energy consumption increment can be easily calculated, which is denoted as p_λ. If $\lambda = 0$ and $d(p_\lambda) \leq \delta_{rel}$, we can also find the optimal solution to the original problem. If $d(p_\lambda) \geq \delta_{rel}$, we must increase λ to strengthen the advantage of availability in the revised cost function. Therefore, we increase λ when the optimal solution of c_λ satisfies the availability requirement.

Based on the Lagrangian relaxation technique, we can obtain the relaxed optimization function as:

$$L(\lambda) := \min\{C_\lambda(p)\} + \lambda\delta_{rel} \quad p \in P(s,t) \tag{13}$$

Theorem 1. *For arbitrary $\lambda \geq 0$, $L(\lambda)$ is a lower bound of Eq. (11).*

Proof. Let p^* denote the optimal solution for Eq. (4), then for $p \in P(s,t)$:

$$\begin{aligned} L(\lambda) &= \min\{C_\lambda(p)\} + \lambda\delta_{rel} \\ &\leq C_\lambda(p^*) + \lambda\delta_{rel} \\ &\leq C(p^*) + \lambda(\delta_{rel} - d(p^*)) \\ &\leq C(p^*). \end{aligned} \tag{14}$$

To derive the optimal lower bound, we need to maximize the function $L(\lambda)$, which means we need to find out a λ^* that can lead Eq. (11) to be maximized£°

$$L^* := \max_{\lambda \geq 0} L(\lambda) \tag{15}$$

Then we give several theorems for function $L(\lambda)$.

Theorem 2. *L is a concave piecewise linear function, that is, for all $p \in P(s,t)$, the minimum value of linear function $C(p)+\lambda(d(p)-\delta_{rel})$.*

Proof. For $p \in P(s,t)$,

$$\begin{aligned} L(\lambda) &= \min\{C(p) - \lambda d(p)\} + \lambda\delta_{rel} \\ &\leq C(p) + \lambda(\delta_{rel} - d(p)). \end{aligned} \tag{16}$$

Theorem 3. *When $\lambda \leq \lambda^*$, $d(p_\lambda) \geq \delta_{rel}$; If $\lambda \geq \lambda^*$, then for any C_λ-minimal path p_λ, $d(p_\lambda) \leq \delta_{rel}$.*

Proof. For C_λ-minimal path p_λ,

$$L(\lambda) = C_\lambda(p_\lambda) + \lambda(\delta_{rel} - d(p_\lambda)). \tag{17}$$

As known in Theorem 2, L is a concave piecewise linear function. When $\lambda \leq \lambda^*$, L has not reached the extreme point and increases as λ is approaching λ^*. $\delta_{rel} - d(p_\lambda) \geq 0$, that is $d(p_\lambda) \geq \delta_{rel}$, increases the value of L. On the other hand, when λ exceeds λ^*, L begins to decrease. As $\lambda > 0$, $\delta_{rel} - d(p_\lambda) \leq 0$, that is $\delta_{rel} \leq d(p_\lambda)$, decreases L. Theoretically, $\delta_{rel} - d(p_\lambda)$ changing from non-positive to positive reflects our dissatisfaction to satisfaction with the constraints of the planning problem.

Theorem 4. *If and only if there exists paths p_c and p_d that satisfy $d(p_c) \geq \delta_{rel}$, $d(p_d) \leq \delta_{rel}$, respectively, and both of them are C_{λ^*}−minimal, then λ maximizes the function $L(\lambda)$. (p_c and p_d can be the same path when $d(p_c) = d(p_d) = \delta_{rel}$.) The algorithm proposed in this paper will give these paths and the value of λ^*.*

Proof. L is a concave piecewise linear function. Only when λ has C_λ-minimal path solutions on the left and right sides of the extreme point λ^*, the maximum L can be obtained. As proved in Theorem 3, when $\lambda \leq \lambda^*$, $d(p_\lambda) \geq \delta_{rel}$; If $\lambda \geq \lambda^*$, then for any C_λ-minimal path p_λ, $d(p_\lambda) \leq \delta_{rel}$. Thus, there are $d(p_c) \geq \delta_{rel}$ and $d(p_d) \leq \delta_{rel}$ on both sizes of the extreme point λ^*.

Theorems above illustrate that the λ^* that maximizes $L(\lambda)$ gives the optimal revised cost function, which means λ^* is the minimum λ value to guarantee the existence of the C_{λ^*}−minimal path p_d that meets the availability requirement.

In general, ignoring constraints and building them into the objective function is the meaning of relaxation technique. Then, the solution to the original problem is of course also applicable to the relaxation condition, so we can thus obtain the lower bound of the original problem. If the found path does not satisfy the constraint, we need to increase its advantage in the revised cost function, reduce the gap between the obtained lower bound and the optimal solution, forcing the obtained solution to approach the optimal solution. This is the basic idea of Lagrangian Relaxation technology [26].

In this paper, we designed a heuristic algorithm based on Lagrangian Relaxation technology to derive the optimal Lagrangian multiplier λ for given node pair. Thus the algorithm can get a near optimal solution compared existing previous resource aggregation based energy-saving routing algorithm.

Algorithm 1: : AEEVP algorithm.

Input: Graph $G(V, L)$, the set of SFC requests F
Output: The path set $S_f, f \in F$
1: Sort SFC requests $f, f \in F$ by their bandwidth
2: **for** each $f \in F$ **do**
3:   Initialize an empty set S_f for storing obtained primary and backup paths
4:   $p_c \leftarrow$ execute function:compute energy increment minimized path
5:   **if** the end-to-end availability $d(p_c) \geq \delta_{rel}$ **then**
6:     add path p_c as primary path to the set S_f
7:   **else**
8:     $p_d \leftarrow$ execute function:compute availability maximized path
9:     **while** $1-(1-d(p_f))(1-d(p_d)) \leq \delta_{rel}$ **do**
10:       $S_f \leftarrow S \bigcup p_d$, add path p_d to the set S_f
11:       $d(p_f) \leftarrow 1-(1-d(p_f))(1-d(p_d))$
12:       $p_d \leftarrow$ execute function:compute availability maximized path
13:     **end while**
14:   **end if**
15:   $p_{cd} \leftarrow$ execute function:Lagrangian Relaxation approach
16:   **return** $S_f \bigcup p_{cd}$
17: **end for**

5.3. Lagrangian Relaxation based heuristic algorithm

In this section, we will present the design and implementation of the Lagrangian Relaxation technology based heuristic algorithm. The algorithm can effectively find the routing path and corresponding set of backup paths while guaranteeing the end-to-end service availability requirement.

AEEVP algorithm: Algorithm 1 first sorts all SFC requests by the bandwidth from high to low. Then the algorithm do the following for each SFC request in set F in turn. Initially, the algorithm allocates an empty set S_f for each SFC request f to store the obtained primary and backup paths (line 1—line 3). Then, the algorithm executes the *energy increment minimized path function*, thus deriving a routing path p_c, so that the energy consumption increment of the whole network caused by deploying the path can be minimized (line 4). Then Algorithm 1 calculates the availability of the mapping path based on the model introduced in the previous section. If the availability $d(p_c)$ satisfies the requirement δ_{rel}, since the backup node and the back path only occupy resources and do not generate additional energy consumption, we can choose p_c as the primary path for the SFC request and add it to the path set S_f (line 5–line 6). Otherwise, the algorithm will repeatedly execute *availability maximized path function* until the end-to-end availability of the SFC request reaches the required δ_{rel}. Each time the algorithm calculates a feasible routing path p_d with current highest availability and add it to path set (line 7–line 14). Then as line 15 shows, the algorithm leverages *Lagrangian Relaxation approach function* for last derived p_d to obtain a path p_{cd} that can not only satisfy the availability requirement, but also is the path with least energy consumption. Then $S_f \bigcup p_{cd}$ is the collection of primary path and backup paths for the SFC request(line 16). The specific implementation of these three functions will be detailed in following parts.

Energy increment minimized path algorithm: As shown in Algorithm 2, the empty set V_{pri} is used to record the location of the VNF deployment node, and the set L_{pri} is used to record the physical link of the mapping path (line 1). Then, for each VNF $i \in f$ in the SFC request f, perform the following operations on them in turn. Define N to record i's the final deployment

Algorithm 2: : Energy increment minimized path algorithm.

Input: $G(V, L)$, SFC request f
Output: The path with the least incremental energy consumption p_c
1: Initialize VNF deployment node set V_{pri} and physical link mapping set L_{pri}
2: **for** each VNF i,$i \in f$ **do**
3: Define N as the final deployment node of i, and initialize the resource utilization of the VNF deployment U_N = -1
4: **for** each network node v,$v \in V$ **do**
5: **if** The resource demand of i is not greater than node v, that is, $Cap_v \geqslant D_i$ **then**
6: **if** The resource utilization of node v is greater than U_N, that is, $U_v > U_N$ **then**
7: $N \leftarrow v$, assign node v to N
8: **end if**
9: **end if**
10: **end for**
11: $V_{pri} \leftarrow V_{pri} \bigcup N$, add node N to the set of master VNF deployment nodes
12: **end for**
13: According to the deployment location of the master VNF, execute the shortest path algorithm to find the path p_c
14: $L_{pri} = l, l \in p_c$
15: **return** p_c,V_{pri},L_{pri}

Algorithm 3: : Availability maximized path algorithm.

Input: $G(V, L) \leftarrow G(V, L - L_{pri})$, SFC request f
Output: The path with the most availability p_d
1: Initialize VNF backup node set V_{bp} and protection link set L_{bp}
2: **for** each VNF i,$i \in f$ **do**
3: Define M as the final protection node of i, and initialize node M's availability, that is $d(M)$=0
4: **for** each network node v,$v \in V - V_{pri}$ **do**
5: **if** the protection resource demand of i is not greater than node v, that is, $Cap_v \geqslant D_i$ **then**
6: **if** the availability of node v is greater than $d(M)$, that is, $d(v) > d(M)$ **then**
7: $M \leftarrow v$, assign node v to M
8: **end if**
9: **end if**
10: **end for**
11: $V_{bp} \leftarrow V_{bp} \cup M$, add node M to the set of protection VNF nodes
12: **end for**
13: According to the location of protection VNFs, execute the Dijkstra algorithm to find the protection path p_d
14: **return** p_d

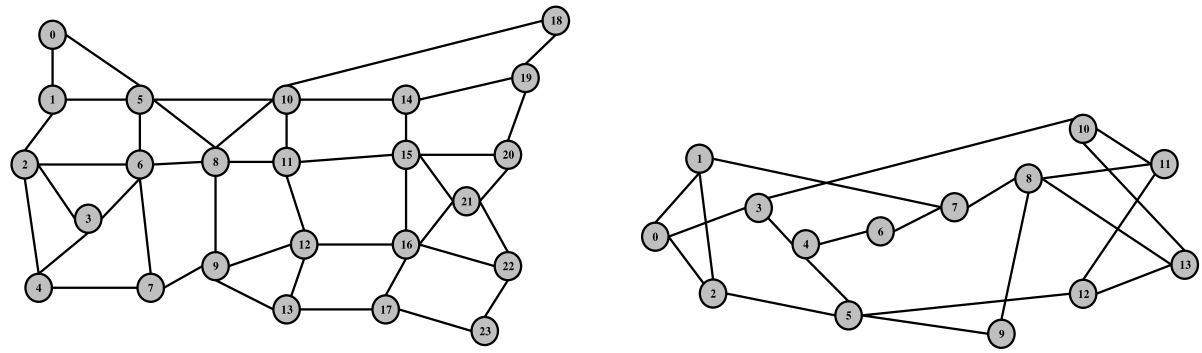
Algorithm 4: : Lagrangian Relaxation approach function.

Input: $G(V, L) \leftarrow G(V, L - L_{pri})$, SFC request f,energy increment minimized path p_c and master VNF deployment node set V_{pri},availability maximized path p_d and backup VNF deployment node set V_{bp}
Output: the low energy consumption path p_{cd} which can meet the availability after joining the path set
1: **while** true **do**
2: $\lambda \leftarrow \frac{Cost(p_c)-Cost(p_d)}{Rel(p_c)-Rel(p_d)}$
3: **for** each VNF i,$i \in f$ **do**
4: **for** each note $n \in V - V_{pri}$ and $Cap_n \geq D_i$ **do**
5: $w_n^i \leftarrow \infty$
6: $V_n^i \leftarrow \emptyset$
7: $j \leftarrow i$'s last VNF
8: **for** each node $m \in V - V_{pri}$,and $Cap_m \geq D_j$ **do**
9: $p_{m,n} \leftarrow$ use the energy consumption increment and availability as the weight of nodes and links in the manner of $Cost + \lambda \cdot Rel$, and the optimal path is obtained by Dijkstra algorithm
10: **if** $w_m^j + \sum_{l \in p_{m,n}} w^l + w_n^i < w_n^i$ **then**
11: $w_n^i \leftarrow w_m^j + \sum_{l \in p_{j,i}} w^l + w_n^i$
12: $p_n^i \leftarrow p_m^j \bigcup p_{m,n}$
13: $V_n^i \leftarrow V_m^j \bigcup n$
14: **end if**
15: **end for**
16: **end for**
17: **end for**
18: According to the path between the found VNF deployment location and VNF, find the path p_λ
19: **if** $Cost(p_d) + \lambda * Rel(p_d) == Cost(p_\lambda) + \lambda * Rel(p_\lambda)$ **then**
20: $p_{cd} \leftarrow p_\lambda$
21: **return** p_{cd}
22: **else**
23: **if** adding p_λ to the path set can meet the availability requirements **then**
24: $p_d \leftarrow p_\lambda$
25: $V_d \leftarrow V_\lambda$
26: **else**
27: $p_c \leftarrow p_\lambda$
28: $V_c \leftarrow V_\lambda$
29: **end if**
30: **end if**
31: **end while**

node location, and initialize the resource utilization at this point to $U_N = -1$ (line 2–line 3). Then traverse each node v in the network that can provide enough computing resources to support VNF i. If the resource utilization $U_v > U_N$ at this point, assign v to N and update N's resource utilization U_N (line 4–line 10). After traversing all nodes, the value of N is the final deployment location of VNF i. After the deployment positions of all VNFs in f are determined, Dijkstra's algorithm is used to calculate the shortest path between VNFs, and finally energy increment minimized path of SFC request f is p_c (line 11–line 15).

Availability maximized path algorithm: First, the physical link L_{pri} of the master path in the network topology is excluded, because the protection scheme does not allow the master path and the protection path to intersect. Then initialize an empty set V_{bp} for the current SFC request, which is used to record the backup VNF node location of the protection path (line 1). For each VNF $i \in f$, perform the following operations. First define M to record the final deployment location of i, and initialize the availability of this point $d(M)$ to 0 (line 2–line 3). Then, traverse each node v in the network that can provide enough computing resources to support the backup VNF i. If the availability of this point $d(v) \geqslant d(M)$, then assign v to M and update the availability of M (line 4–line 10). After traversing, add the M as the final deployment location of i to the backup VNF deployment location set V_{bp} in line 11. Finally, based on the calculated deployment location of all backup VNFs, using the availability of the physical link as the weight, the shortest path between VNFs is calculated by the Dijkstra algorithm, and the complete backup path p_d with the highest availability is obtained as well (line 13–line 14).

Lagrangian Relaxation approach function: Algorithm 4 firstly excludes the physical link L_{pri} of the master path in the network topology, because the protection scheme does not allow the master path and the protection path to intersect. For SFC request



(a) USNET topology (b) NSFNET topology

Fig. 5. Simulation topology.

f, after determining the mapping node and the corresponding path, the energy consumption increase generated by each node and each physical link after mapping this request can be easily calculated (nodes without VNF deployment do not increase energy consumption), expressed as $Cost(v)$, $Cost(l)$, $v, l \in V, L$, while $Cost(p)$ is used to represent the total energy cost of this path. The availability of the path can also be calculated based on the availability of the nodes and links on this path, and $Rel(p) = -\ln^{d(p)}$ is used to represent the reliability cost of the path. Then calculate the value of Lagrange multiplier λ according to the entered p_c, p_d and Eq. (7) (line 2).

$$\lambda \leftarrow \frac{Cost(p_c) - Cost(p_d)}{Rel(p_c) - Rel(p_d)}, \tag{18}$$

Then for each VNF $i \in f$, execute the following steps. Traverse all nodes in the network that meet the requirements of i computing resources among $n \in V - V_{pri}$, firstly initialize a weight value to the node as w_n^i, indicating the weight of VNF i deployed on node n, then initialize an empty set V_n^i to record VNF i and previous VNF mapping nodes. Let j denote the last VNF node of i and traverse all the nodes $m \in V - V_{pri}$ in the network that meet the computing resources requirement of j, and integrate the energy consumption increment and availability as the weight of the nodes and links in the manner of $Cost + \lambda \cdot Rel$. The best weight path $p_{m,n}$ can be obtained by using the improved Dijkstra algorithm in Ref. [27] (line 4–line 9). During the traversal comparison process, the three parameters of $p_{m,n}$, w_n^i, and V_n^i are continuously updated (line 10–line 13). Finally, the path p_λ is obtained according to the location of the VNF deployment and the path between the VNFs in line 18. If $Cost(p_d) + \lambda * Rel(p_d) == Cost(p_\lambda) + \lambda * Rel(p_\lambda)$, assign p_λ to p_{cd} and end the algorithm (line 19–21). Otherwise, update the value of p_c or p_d in two cases, and repeat the above steps. If adding p_λ to the path set is enough to meet the availability requirement δ_{rel}, then p_λ is assigned to p_d (line 22–line 25); otherwise, p_λ is assigned to p_c (line 26–line 31).

5.4. Time complexity analysis

AEEVP algorithm is based on the Lagrangian Relaxation approach function, while Lagrangian Relaxation approach function is derived from Energy increment minimized path algorithm and Availability maximized path algorithm. Since the time complexity of *Dijkstra* algorithm is $O(|N|^2)$, where $|N|$ indicates the number of nodes in the graph G, the time complexity of Energy increment minimized path algorithm and Availability maximized path algorithm is $O(|V||N|^2)$, where $|V|$ indicates the number of VNFs required for the service request r. In our simulation model, the number of VNFs $|V|$ is much smaller than the number of nodes $|N|$ in map G. As a result, the time complexity of Energy increment minimized path algorithm and Availability maximized path algorithm can be expressed as $O(|N|^4)$. In the Lagrangian Relaxation approach function, we can see two loop iterations of nodes in map G, so the time complexity of is $O(|N|^4)$. Finally, the time complexity of AEEVP algorithm is $O(|F||V|^4)$, where $|F|$ represents the number of service requests.

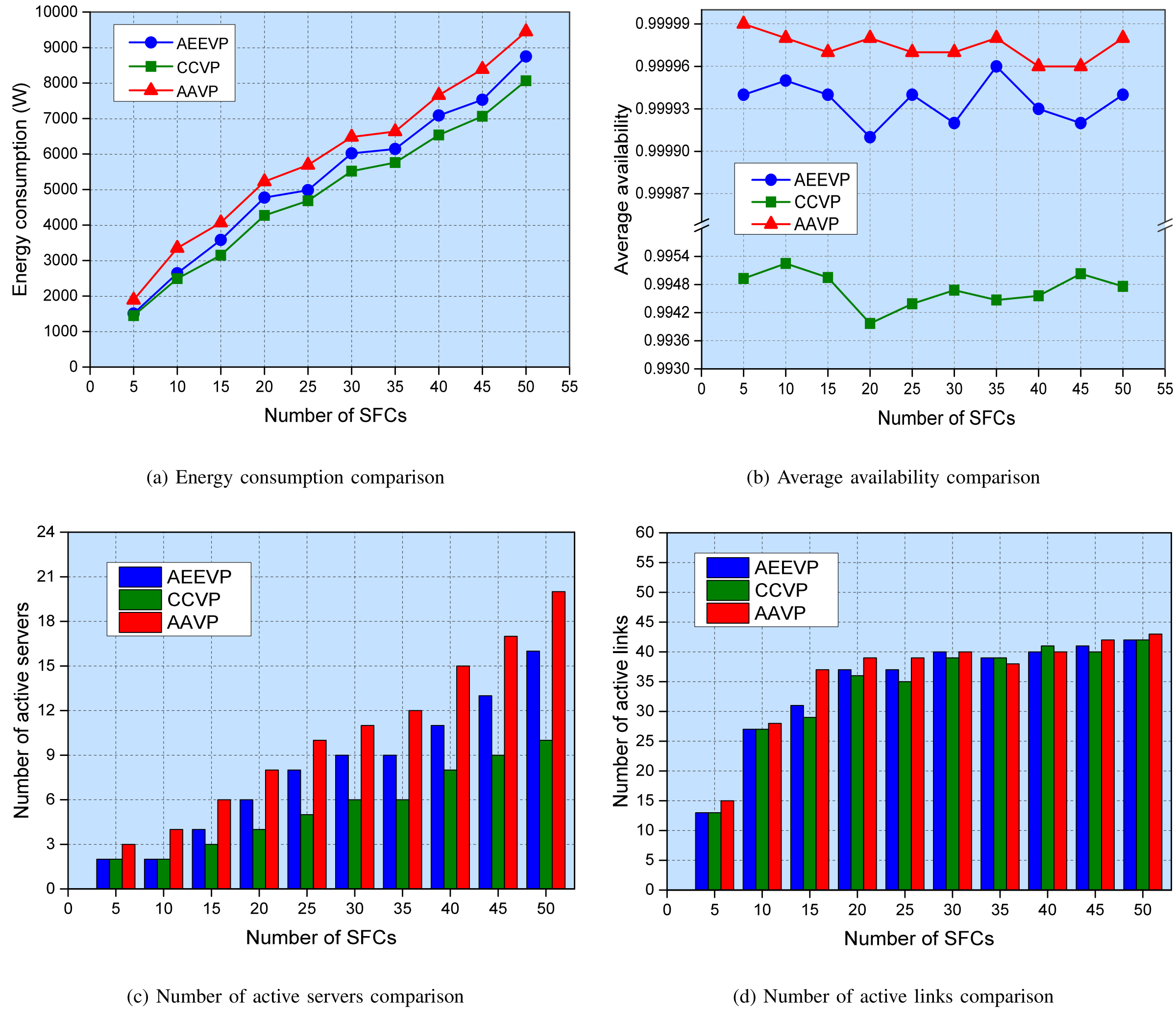
6. Performance evaluation

This section presents the simulation results of the performance evaluation. In simulation, we test on two typical telecom network topology, which are shown in Fig. 5, including USNET topology (24 nodes and 43 bidirectional links) and NSFNET topology (14 nodes and 21 bidirectional links). Due to the limitation of topology size, we assume there are three types of VNF (i.e., Firewall, IDS, and WAN-opt) to be deployed on the servers. Each server is characterized by its: (1) computing capacity, which is set as 500 Mbps; (2) availability, which is set to be uniformly distributed in the range of [0.9990, 0.9999]. We set the capacity of each link as 1000 Mbps and the availability is also set to be uniformly distributed in the range of [0.9990, 0.9999]. To simulate the actual network traffic, the SFCs are set to be uniformly distributed among all node pairs and the number of VNFs for each chain is uniformly distributed in the range of [1, 3]. We assume that the bandwidth of SFCs is uniformly distributed in the range of [40 Mbps, 50 Mbps]. The availability requirement of SFC is set as 0.9999. Generally speaking, the energy consumption of server is higher than that of link. Thus we set the startup and peak-load energy consumption of each server as 170 W and 500 W, respectively. The startup and peak-load energy consumption of each link is set as 50 W and 200 W, respectively.

In our simulation study, we use the following four performance metrics to evaluate our algorithm:

1. Energy consumption: the total amount of energy consumption of the network, which consists of energy consumption of all servers and physical links;
2. Average availability: the average availability of all the SFCs;
3. Number of active servers: the number of servers activated for hosting VNFs;
4. Number of active links: the number of links activated for carrying services traffic.

We compare our proposed AEEVP algorithm with energy-efficient only algorithm which is based on cost-efficient centrality-based VNF placement and chaining (CCVP) algorithm [28], and availability-aware only (AAVP) algorithm.



(a) Energy consumption comparison

(b) Average availability comparison

(c) Number of active servers comparison

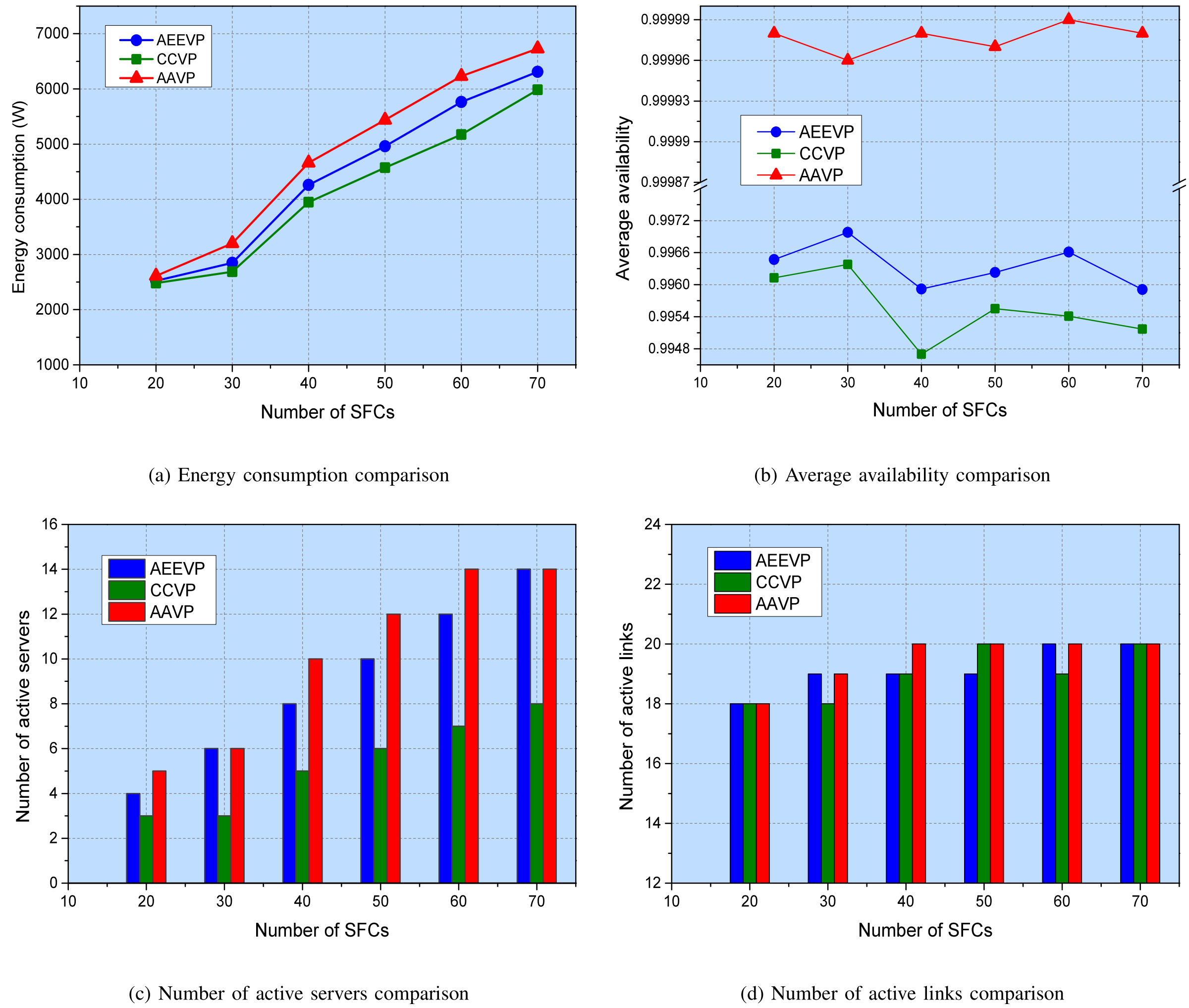
(d) Number of active links comparison

Fig. 6. Simulation results of USNET topology.

Fig. 6(a) shows the energy consumption of the three algorithms in the USNET topology. We can see that the energy consumption increases with the increase of SFCs. The CCVP algorithm consumes the fewest energy consumption, our AEEVP algorithm performs second best, while the AAVP algorithm is shown to be the largest. The reason is that the CCVP algorithm consolidates VNFs and services in the network for the purpose of energy efficiency. The AAVP algorithm provides backup VNFs and backup paths to enhance end-to-end service availability, which leads to that more network devices needed to be started up for providing equal services and thus consumes more energy consumption. Our AEEVP algorithm guarantees service availability by adding the minimal set of backup paths, and cuts down energy consumption by leveraging LARAC algorithm while guaranteeing service availability. Thus, the AEEVP algorithm can balance the two above factors.

Fig. 6(b) shows the average availability of the three algorithms in the USNET topology. The number of SFC requests is increased gradually from 5 to 50 by 5 each time. It can be seen that the average reliability of the three algorithms has no obvious relationship with the number of SFC requests. We can see that our AEEVP algorithm and the AAVP algorithm both guarantee the end-to-end service availability requirement (i.e., 0.9999), while the average availability of the CCVP algorithm is obviously lower than the requirement. In detail, we can also see that the average availability of our AEEVP algorithm is slightly lower than that of the AAVP algorithm. The reason is that our AEEVP algorithm exchanges service availability with energy efficiency by minimizing the number of backup paths while the AAVP algorithm only aims at the optimization of end-to-end service availability. In addition, since the CCVP algorithm does not consider the end-to-end service reliability, its reliability is much lower than the reliability requirements of the other two algorithms.

Fig. 6(c) and Fig. 6(d) shows the number of active servers and links in the three algorithms of the USNET topology, respectively. In our simulation, network devices with no workload do not consume any energy, i.e, a device can be turned into sleep mode if it is just used for backup. When a failure occurs in the network, traffic will be switched to the corresponding backup device, and the device will generate energy consumption at this time. We can see that active servers and active links both increase with the increase of SFCs. To be specific, the AAVP algorithm uses more active servers and links than the other two algorithms. The reason is that the other two algorithms both leverage the traffic aggregation method to save energy. In addition, our AEEVP algorithm uses more active servers and links than the CCVP algorithm. The reason is that although the backup VNF and backup routing path do not generate energy consumption under normal circumstances, in order to ensure that the same amount of network



(a) Energy consumption comparison

(b) Average availability comparison

(c) Number of active servers comparison

(d) Number of active links comparison

Fig. 7. Simulation results of NSFNET topology.

services can be supported when a failure occurs, resources need to be reserved, they also occupy computing and communication resources. Therefore, additional servers and physical links need to be activated to provide services for the same amount of SFCs.

We also tested the performance of the above three algorithms in four aspects in the NSFNET topology, and the results are consistent with the results of the USNET topology. As we can see in Fig. 7, with the increase of the SFC number, all four indicators, which are energy consumption, average availability, active servers and links, obviously are increasing under three algorithms. Fig. 7(a) shows the energy consumption of three algorithms in the NSFNET topology. The increase of the AEEVP algorithm is between the AAVP algorithm and the CCVP algorithm. Fig. 7(b) shows the comparisons of average availability with the three algorithms. The AEEVP algorithm performs worse than AAVP algorithm but better than the CCVP algorithm. It means the AEEVP algorithm balances the energy consumption and average availability as it is shown in Fig. 6(a) and Fig. 6(b). As for the active servers and links used in the three algorithms, Fig. 7(c) and Fig. 7(d) shows the AEEVP consumes more computing and communication resources than the CCVP algorithms since back up servers and links are enabled to guarantee the availability. While it still saves some energy consumption than the AAVP algorithm by leveraging LARAC algorithm. Thus, the AEEVP algorithm balances the energy consumption and the availability as it performs in the USNET topology.

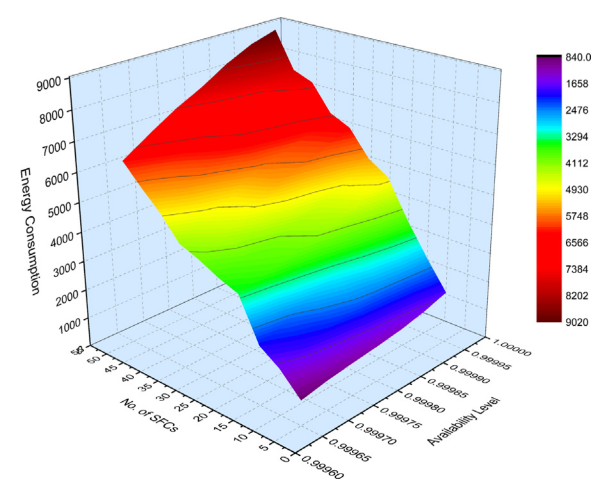


Fig. 8. Trade-off between energy efficiency and end-to-end service availability in our AEEVP algorithm of USNET topology.

Fig. 8 reveals the relationship between energy efficiency and service availability in our AEEVP algorithm of the USNET topology. In this scenario, we vary the number of SFCs in the range of [5, 50] and the service availability requirement in the range of [0.9996, 0.99995]. From this figure, we can see that, with the increase of SFCs and service availability requirement, the energy consumption increases. It means that, to meet the end-to-end service availability requirement, our AEEVP algorithm sacrifices energy efficiency to trade for higher end-to-end service availability. In detail, to achieve increasing end-to-end service reliability requirements, the AEEVP algorithm obtains reliability improvements by increasing the backup VNFs and backup paths. This behavior will take up more network resources, leading to more network equipment must be turned on, resulting in more energy consumption. The reason is that some redundant resources are reserved for backup VNFs and more devices need to be started up to serve the same amount of SFC requests. In this way, the end-to-end service availability requirement can be guaranteed. In summary, our AEEVP algorithm achieves energy minimization while guaranteeing service availability in NFV-enabled networks.

7. Conclusion and future work

In this paper, we have investigated the trade-off between energy efficiency and service availability for NFV. We first present our designed green orchestration NFV architecture. Then we illustrate the energy model and service availability model for NFV-enabled networks. Afterwards, we present a novel energy efficient design algorithm leveraging the off-site backup method to guarantee end-to-end service availability. Finally, simulation results have demonstrated the effectiveness of our proposed algorithm.

Many possible extensions can be designed in the research field described in this paper, and many new research approaches can be developed. First of all, the basic idea of energy efficient design for communication networks is aggregating traffic on fewer nodes and links and minimizing the number of active network devices. It can be considered as a static optimization problem given a set of network services. However, in the networking scenarios of NFV (including cloud data centers, wide area networks, edge computing, etc.), network services always randomly arrive and depart in NFV-enabled networks. Therefore, how to make the energy consumption adapt to dynamic network services is an important issue to study. Secondly, it is clear that energy efficiency is not the only objective for NFV. Other network performances, such as delay, traffic throughput and load balancing, are also need to be carefully taken into consideration. Therefore, how to achieve a trade-off between such conflicting objectives is an interesting problem for NFV energy-efficient design.

CRediT authorship contribution statement

Liuyang Mai: Algorithm design, Writing - original draft, Experiment. **Yi Ding:** Mathematical modeling and Experiment. **Xiaoning Zhang:** Conceptualization, Methodology. **Lang Fan:** Writing - review & editing. **Shui Yu:** Writing - review & editing. **Zhichao Xu:** Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by National Natural Science Foundation of China (NSFC) under Grant 61671124 and 61871097.

References

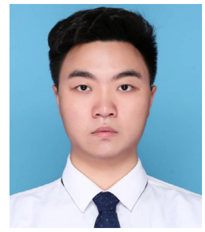
[1] B. Han, V. Gopalakrishnan, L. Ji, S. Lee, Network function virtualization: challenges and opportunities for innovations, IEEE Commun. Mag. 53 (2) (2015) 90–97.
[2] R. Mijumbi, J. Serrat, J. Gorricho, N. Bouten, F. Turck, R. Boutaba, Network function virtualization: state-of-the-art and research challenges, IEEE Commun. Surv. Tutor. 18 (1) (2016) 236–262.
[3] M. Mechtri, C. Ghribi, D. Zeghlache, A scalable algorithm for the placement of service function chains, IEEE Trans. Netw. Serv. Manag. 13 (3) (2016) 533–546.
[4] R. Bolla, R. Bruschi, F. Davoli, L. Gregorio, P. Donadio, L. Fialho, M. Collier, A. Lombardo, D. Recupero, T. Szemethy, Green abstraction layer (gal): power management capabilities of the future energy telecommunication fixed network nodes, ETSI Technical Report, 2014.
[5] N. ISG, Network functions virtualisation (nfv)-virtualisation requirements, ETSI Technical Report, 2013.
[6] M. Wang, B. Cheng, S. Zhao, B. Li, J. Chen, Availability-aware service chain composition and mapping in NFV-enabled networks, in: Proc. IEEE International Conference on Web Services (ICWS), 2019.
[7] J. Fan, C. Guan, Y. Zhao, C. Qiao, Availability-aware mapping of service function chains, in: Proc. IEEE International Conference on Computer Communications (INFOCOM), 2017.
[8] P. Grandelag, Energy-efficient cooling for telecom networks: thermosiphon as energy savings generator, in: Proc. IEEE International Telecommunications Energy Conference (INTELEC), 2015.
[9] Z. Abulfazl, N. Gholipoor, M.R. Javan, N. Mokari, Energy cost minimization by joint radio and NFV resource allocation: E2E QoS framework, IEEE Trans. Wirel. Commun. (2019) Early Access.
[10] O. Soualah, M. Mechtri, C. Ghribi, D. Zeghlache, Energy efficient algorithm for VNF placement and chaining, in: Proc. IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID), 2017.
[11] C. Pham, N.H. Tran, S. Ren, W. Saad, C.S. Hong, Traffic-aware and energy-efficient VNF placement for service chaining: joint sampling and matching approach, IEEE Trans. Serv. Comput. 13 (1) (2020) 172–185.
[12] S. Ahvar, H.P. Phyu, S.M. Buddhacharya, E. Ahvar, N. Crespi, R. Glitho, CCVP: cost-efficient centrality-based VNF placement and chaining algorithm for network service provisioning, in: Proc. IEEE Conference on Network Softwarization (NetSoft), 2017.
[13] C. Tipantuna, X. Hesselbach, NFV/SDN enabled architecture for efficient adaptive management of renewable and non-renewable energy, IEEE Open J. Commun. Soc. 1 (2020) 357–380.
[14] X. Zhong, Y. Wang, X. Qiu, Cost-aware service function chaining with reliability guarantees in NFV-enabled inter-DC network, in: Proc. IFIP/IEEE Symposium on Integrated Network and Service Management (IM), 2019.
[15] H.R. Khezri, P.A. Moghadam, M.K. Farshbafan, V. Shah-Mansouri, H. Kebriaei, D. Niyato, Deep reinforcement learning for dynamic reliability aware NFV-based service provisioning, in: Proc. IEEE Global Communications Conference (GLOBECOM), 2019.
[16] W. Ding, H. Yu, S. Luo, Enhancing the reliability of services in NFV with the cost-efficient redundancy scheme, in: Proc. IEEE International Conference on Communications(ICC), 2017.
[17] M.A.T. Nejad, S. Parsaeefard, M.A. Maddah-Ali, T. Mahmoodi, B.H. Khalaj, Vspace: VNF simultaneous placement, admission control and embedding, IEEE J. Sel. Areas Commun. 36 (3) (2018) 542–557.
[18] S. Yang, F. Li, R. Yahyapour, Delay-sensitive and availability-aware virtual network function scheduling for NFV, IEEE Trans. Serv. Comput. (2019) Early Access.
[19] P. Torkamandi, S. Khorsandi, B. Bakhshi, Availability aware SFC embedding in NFV: a clustering approach, in: Proc. Iranian Conference on Electrical Engineering (ICEE), 2019.
[20] ETSI GS NFV-MAN 001, Network functions virtualisation (NFV); management and orchestration, 2014.
[21] M. Pedram, I. Hwang, Power and performance modeling in a virtualized server system, in: Proc. IEEE International Conference on Parallel Processing Workshops, 2010.
[22] K. Zheng, X. Wang, L. Li, X. Wang, Joint power optimization of data center network and servers with correlation analysis, in: Proc. IEEE International Conference on Computer Communications (INFOCOM), 2014.
[23] A. Hmaity, M. Savi, F. Musumeci, M. Tornatore, A. Pattavina, Virtual network function placement for resilient service chain provisioning, in: Proc. International Workshop on Resilient Networks Design and Modeling (RNDM), 2016.

[24] M. Held, R.M. Karp, The traveling-salesman problem and minimum spanning trees, Math. Program. 18 (1970) 1138–1162.
[25] M. Held, R.M. Karp, The traveling-salesman problem and minimum spanning trees: Part II, Math. Program. 1 (1971) 6–25.
[26] R.K. Ahuja, T.L. Magnanti, J.B. Orlin, Network Flows, prentice hall, 1988.
[27] C. Yin, H. Wang, Developed dijkstra shortest path search algorithm and simulation, Int. Conf. Comput. Des. Appl. 1 (2010) 116–119.
[28] S. Ahvar, H.P. Phy, S.M. Buddhacharya, E. Ahvar, R. Glitho, CCVP: cost-efficient centrality-based VNF placement and chaining algorithm for network service provisioning, in: Proc. IEEE Network Softwarization, 2017.

Liuyang Mai is currently pursuing the Master degree in communication and information system with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, China. Her research interests include network optimization, software defined networks, and network function virtualization.

Yi Ding is currently pursuing the Master degree in electronic and communication engineering with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, China. His research interests include edge computing, software defined networks, and network function virtualization.

Xiaoning Zhang received the B.S., M.S., and Ph.D. degrees in Information and Communication engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2002, 2005, and 2007, respectively. He is currently an Professor with School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, China. His research interests include network design, software defined networks and network function virtualization.



Lang Fan is currently pursuing the Ph.D. degree in information and communication engineering with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, China. His research interests include edge computing, software defined networks, and network function virtualization.

Shui Yu received his B.Eng. (Electronic Engineering) and M.Eng. (Computer Science) degree from University of Electronic Science and Technology of China, P.R. China in 1993 and 1999, respectively. He also obtained an Associate Degree in Mathematics from the same university in 1993. He received his Ph.D. (Computer Science) from Deakin University in 2004. He is currently a full Professor of School of Software, University of Technology Sydney, Australia. His research interest includes Security and Privacy, Networking, Big Data, and Mathematical Modelling. He has published two monographs and edited two books, more than 200 technical papers, including top journals and top conferences, such as IEEE TPDS, TC, TIFS, TMC, TKDE, TETC, ToN, and INFOCOM. Dr Yu initiated the research field of networking for big data in 2013. His h-index is 33. Dr Yu actively serves his research communities in various roles. He is currently serving the editorial boards of IEEE Communications Surveys and Tutorials, IEEE Communications Magazine, IEEE Internet of Things Journal, IEEE Communications Letters, IEEE Access, and IEEE Transactions on Computational Social Systems. He has served many international conferences as a member of organizing committee, such as publication chair for IEEE Globecom 2015, IEEE INFOCOM 2016 and 2017, TPC chair for IEEE BigDataService 2015, and general chair for ACSW 2017. Dr Yu is a final voting member for a few NSF China programs in 2017. He is a Senior Member of IEEE, a member of AAAS and ACM, the Vice Chair of Technical Committee on Big Data of IEEE Communication Society, and a Distinguished Lecturer of IEEE Communication Society.

Zhichao Xu received the Master degree in Information and Communication engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2019. His research interests include software defined networks and network function virtualization.