Location-Aware Resource Allocation Algorithm in Satellite Ground Station Networks

Xiangqiang Gao, Rongke Liu, *Senior Member, IEEE* and Aryan Kaushik, *Member, IEEE*

Abstract—As per the increase in satellite number and variety, satellite ground station should be required to offer user services in a flexible and efficient manner. Network function virtualization (NFV) can provide a new paradigm to allocate network resources on demand for user services over the underlying network. In this paper, we investigate the virtualized network function (VNF) placement and routing traffic problem in satellite ground station networks. We formulate the problem of resource allocation as an integer linear programming (ILP) model and the objective is to minimize the link resource utilization and the number of servers used. Considering the information about satellite orbit fixation and mission planning, we propose location-aware resource allocation (LARA) algorithms based on Greedy and IBM CPLEX 12.10, respectively. The proposed LARA algorithm can assist in deploying VNFs and routing traffic flows by predicting the running conditions of user services. We evaluate the performance of our proposed LARA algorithm in three networks of Fat-Tree, BCube and VL2. Simulation results show that our proposed LARA algorithm performs better than that without prediction, and can effectively decrease the average resource utilization of satellite ground station networks.

Index Terms—Network function virtualization (NFV), satellite ground station, resource allocation, resource utilization, greedy algorithm, IBM CPLEX.

1 Introduction

Software Defined Network (SDN) [1] and Network Function Virtualization (NFV) [2] play an important role in data center networks [3]. They can implement the separation of module functions and dedicated hardware equipments, where the module functions are referred to as Virtualized Network Functions (VNFs) and run on commodity servers [4], [5]. In general, a user service is considered as a service function chaining (SFC) which consists of several VNFs and is represented as a directed acyclic graph (DAG), where traffic flows in networks need to pass through the VNFs in a specific order [6]. Within physical network resource constraints, network service provider can flexibly place VNFs on network nodes and decide routing paths for traffic flows to optimize the operating efficiency in terms of energy consumption, resource utilization, operation cost etc. [4], [6], [7], [8]. As a new paradigm, the two technologies have a profound influence on the next generation networks [9].

The conventional satellite ground station (SGS), which consists of expensive dedicated hardwares, is more complicated and difficult to be compatible with different user services, as the number and variety of services increases. SGS networks can effectively improve the performance of resource allocation by introducing SDN and NFV approaches [10], [11], [12]. However, most of previous work about VNF placement and routing traffic focuses on improving system models and optimizing resource allocation algorithms in data centers, enterprise networks, cloud computing etc. [7], [13], [14], [15]. There are a few related work about studying the problem of resource allocation in SGS networks [16], [17], [18], [19].

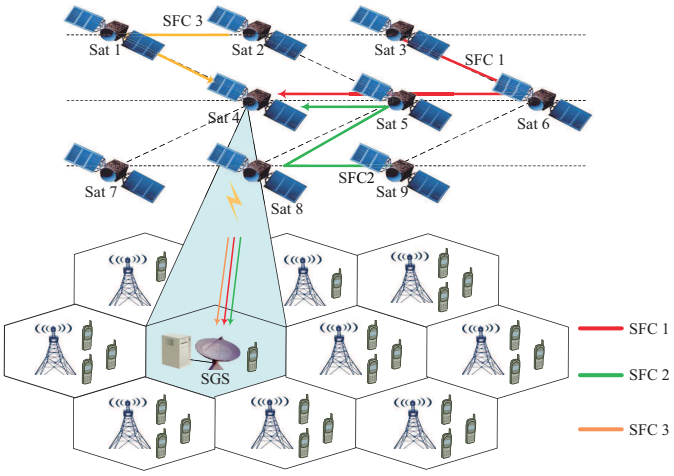


Fig. 1. Procedure of running user services.

For satellite communication systems, as the results of satellite physical resource constraints, the number of payloads carried by a satellite is limited and each satellite has a fixed orbit [20]. To effectively provide satellite services for users, satellite control center operates the satellite mission planning and orchestrate user services in the light of service requirements and satellite resources conditions [21], [22], [23]. According to the results of the satellite mission planning in satellite control center, satellite ground station can prior obtain the information concerning running user services, which includes service type, resource requirements and service start and end time, and effectively deploy VNFs and route traffics. Fig. 1 describes the procedure of running

- *X. Gao and R. Liu are with the School of Electronic and Information Engineering, Beihang University, Beijing, China. E-mail: {xggao, rongke_liu}@buaa.edu.cn.*
- *Aryan Kaushik is with Department of Electronic and Electrical Engineering, University College London (UCL), London, United Kingdom. E-mail: a.kaushik@ucl.ac.uk.*

three user services in satellite communication networks. The traffic flows are Sat3 → Sat6 → Sat5 → Sat4 → SGS for SFC1, Sat9 → Sat8 → Sat4 → SGS for SFC2 and Sat2 → Sat1 → Sat4 → SGS for SFC3, respectively. SGS can be more efficient to assign the network resources to the three user services by acquiring the information of the satellite mission planning in advance.

In this paper, we study the problem of VNF placement and routing traffic in SGS networks. An Integer Linear Programming (ILP) model is formulated to minimize the link resource utilization and the number of servers used. To address the optimization problem, we propose location-aware resource allocation (LARA) algorithms based on Greedy [24] and IBM CPLEX 12.10 [25], respectively, according to predicting the running conditions of user services by the satellite mission planning. Note that the satellite mission planning for user services is out the scope of this paper and we assume all the information about the satellite mission planning of user services can be known in advance, which includes service type, resource requirements and the start and end time of services. We make the experiments for three networks of Fat-Tree [26], Bcube [27] and LV2 [28] with different number of servers to evaluate the performance of our proposed LARA algorithm. This paper provides the following contributions.

- We build the problem of VNF placement and routing traffic by prior sensing the running conditions of satellite user services in SGS networks, where the information about resource requirements, service type and the life cycle time for all user services could be predicted via the satellite mission planning in satellite control center.
- We formulate the problem of VNF placement and routing traffic as an ILP model and prove it to be NP-hard. Our aim is to minimize the resource utilization of networks.
- Two location-aware resource allocation algorithms based on Greedy and CPLEX are implemented to address the problem of VNF placement and routing traffic.
- We evaluate the performance of our proposed Greedy- and CPLEX-based LARA algorithms in BCube networks with 4 and 8 servers, respectively, and can observe that the proposed LARA algorithm based on CPLEX is suitable for solving the problem of resource allocation in small scale networks due to the computational complexity.
- Furthermore, we simulate and evaluate the performance of our proposed Greedy-based LARA algorithm for different number of predictable time slots in three networks of Fat-Tree, BCube and LV2 with 16 servers.

The remainder of this paper is organized as follows: Section 2 briefly reviews related work about VNF placement and routing traffic problem. Section 3 introduces the system model of resource allocation in terms of physical network and user services. In Section 4, we formulate the problem of resource allocation as an ILP model and analyze the computational complexity. Location-aware resource allocation algorithms are proposed based on Greedy and IBM CPLEX in Section 5. Section 6 discusses the performance of our proposed LARA algorithm in three different networks. Finally, we provide the conclusion of this paper in Section 7.

2 Literature Review

The problem of VNF placement and routing traffic in cloud environment is demonstrated as NP-hard [8], [24]. Most of existing literature [29], [30], [31], [32] focuses on managing network resources and improving resource allocation algorithms to optimize their objective functions, e.g., minimizing energy cost, maximizing resource utilization and improving quality of service (QoS). Due to the computational complexity of the ILP problem, heuristic algorithms are used to find an approximated solution in practical applications [7], [24], [33]. Some of existing work discuss that resource and workload prediction assists in improving the operation efficiency of network [34], [35], [36].

A fast elitist non-dominated sorting genetic algorithm (NSGA-II) is studied to allocate service resources in cloud, where the authors just considered the resource allocation for virtual machines and routing traffic problem was not discussed in [33]. The authors in [29] presented VNF placement for SFCs to minimize energy consumption and addressed them with CPLEX. In [8], a VNF orchestration problem is discussed to minimize the operational expenditure of network and resource fragment. The authors assumed that some VNFs can only run on a particular set of servers and several SFCs can share a VNF instance. VNF placement with replications is presented in [37] to assist in load balance, where VNFs for a SFC can be replicated according to actual resource requirements of SFCs in networks. In this paper, we formulate the ILP problem to optimize the resource utilization in terms of servers, bandwidths and links, where our work considers that VNFs can be deployed on any servers and a VNF instance can only be used by a SFC.

In [24], the authors proposed two heuristic algorithms based on greedy and simulated annealing to minimize the end-to-end delay and the bandwidth consumption. The authors in [7] formulated an ILP problem to optimize resource utilization of servers, links and bandwidths, and used genetic algorithm to address the resource allocation issue. However, the prediction of SFCs was not considered. In our work, the running conditions of user services can be predicted by the satellite mission planning in satellite control center and we implement the location-aware resource allocation algorithms based on Greedy and IBM CPLEX to address this problem of resource allocation.

A forecast-assisted SFCs placement by affiliation-aware VNF placement is presented in [34], where the future VNF requirements can be forecasted based on fourier-series prediction method. In [35], the authors proposed a traffic forecasting method by analyzing the traffic characteristics in data center networks and implemented two VNF placement algorithms to scale the VNF instances dynamically, where the optimization problem is formulated to minimize the number of virtual machines for deploying VNFs. Deep-learning-assisted VNF orchestration in data center elastic optical network is discussed in [36]. The authors built a deep-learning model by memory-based neural network to

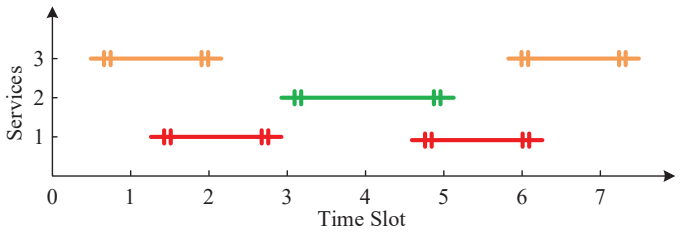


Fig. 2. life cycle time for user services.

predict the future SFCs in pre-deployment phase and implemented an effective training scheme. In our work, we assume that the information about service type, resource demands and life cycle time for all user services can be prior known depending on the satellite mission planning in satellite control center, and propose the LARA algorithm to address the problem of resource allocation.

In some of previous work [38], [39], [40], SDN and NFV are introduced into satellite communication to facilitate the flexibility and scalability. In [39], [40], an innovative architecture of satellite ground systems is discussed by using SDN and NFV to provide satellite communication services. An optimal resource allocation for satellite ground segment system is discussed to effectively orchestrate satellite network services in [41]. A shared satellite ground station is proposed by using user-oriented virtualization to address complex satellite telemetry, tracking and command (TT&C) in [42]. References [17] and [16] propose neighbor-area and tabu search algorithms to solve task scheduling of satellite ground stations, respectively.

To the best of our knowledge, the issue of orchestrating satellite user services for satellite ground stations by predicting the running condition of user services has not been studied. The problem of VNF placement and routing traffic by prior sensing the running conditions of user services would be useful to be investigated in SGS networks.

3 System Model

In this section, we describe the system model for user services and satellite ground station networks in detail, and discuss the problem of network resource allocation for user services, where SGS network and user services are considered as directed acyclic graphs (DAGs).

3.1 User Service

We denote the set of user services as Q, which includes K user services. Each user service $q_k \in Q$ is seen as a service function chaining and can be expressed as a directed acyclic graph $G(F_k, H_k)$. $F_k = f_{k,1}, f_{k,2}, \cdots, s_k, d_k$ denotes the VNFs in q_k, where s_k and d_k indicate the ingress and egress, respectively, and $f_{k,i}$ indicates the i-th VNF of q_k. H_k denotes the set of edges and each edge $h_k^{i_1,i_2} \in H_k$ indicates that there is a bandwidth demand $b_k^{i_1,i_2}$ between f_{k,i_1} and f_{k,i_2}. Note that we assume that there can be various bandwidth demands for different edges. The r-th resource requirements of $f_{k,i}$ are denoted as $c_{k,i}^r$. We assume that s_k and d_k just route traffic flows over networks, and are not required for any computing and storage resources of servers.

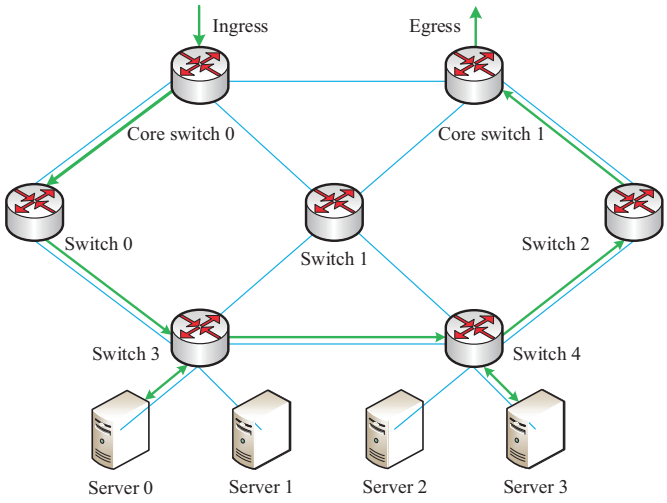


Fig. 3. Data center network.

In addition, satellite networks can provide service for each user during a specific running period, where the start and end time for a user service is fixed. In Fig. 2, an example of the running periods for three user services is shown. It can be observed that each user service has a specific running period and can be carried out by satellites over time slots variation. We denote the running period for user service q_k as $t_{k,p}$. Depending on the satellite mission planning in satellite control center, we assume that the service type, resource requirements, and life cycle time for all user services can be prior obtained.

3.2 Physical Network

Let us denote the underlying physical network as a directed graph $G(V, E)$, where V represents the set of network nodes, including servers, core switches, aggregation and edge switches, and E represents the set of all links where L_e is the total number of links. We denote the set of servers, where the total number of servers is N_{svr} as V_{svr}, the set of core switches as V_{cs} and the set of servers and core switches as V_s. The variable R indicates the set of resources supported by servers, e.g., central processing unit (CPU), memory and graphics processing unit (GPU). The variable C_n^r is the capacity of the r-th resource for the n-th server. We assume that there are two links (v_i, v_j) and (v_j, v_i) between any two adjacent nodes $v_i \in V$ and $v_j \in V$. Let us denote the bandwidth capacity of the l-th link as B_l. Fig. 3 shows the architecture of a data center network. There are four servers, two core switches and five aggregation and edge switches. Different network nodes are connected with bidirectional links. A user service is running on the network. The traffic flows are described as: Core switch0 $\rightarrow$ Switch0 $\rightarrow$ Switch3 $\rightarrow$ Server0 $\rightarrow$ Switch3 $\rightarrow$ Switch4 $\rightarrow$ Server3 $\rightarrow$ Switch4 $\rightarrow$ Switch2 $\rightarrow$ Core switch1.

3.3 Location-Aware Resource Allocation Problem

In this paper, based on the satellite mission planning in satellite control center, we assume that service type, resource requirements and running period time for all user services

TABLE 1
List of Symbols

Physical Network	
V	Set of servers and all switches in network.
V_{svr}	Set of servers with the number of N_{svr} in network.
V_{cs}	Set of core switches in network.
V_s	Set of core switches and servers in network.
E	Set of L_e links in network.
B_l	Bandwidth capacity of the l-th link.
R	Set of resources supported by servers.
C_n^r	Capacity of the r-th resource for the n-th server node.
P_{n_1,n_2}	Set of the shortest d paths between v_{n_1} and v_{n_2}.
P	Set of all paths from each pair of source and destination.
Requested Services	
Q	Set of user services with the number of K.
q_k	The k-th user service.
F_k	Set of virtual network functions (VNFs) offered by q_k.
H_k	Set of edges from q_k.
$h_k^{i_1,i_2}$	Edge between f_{k,i_1} and f_{k,i_1}.
$f_{k,i}$	The i-th vnf of k-th user service.
s_k, d_k	Source and destination of the k-th user service.
$c_{k,i}^r$	The r-th resource requirements for $f_{k,i}$.
$b_k^{i_1,i_2}$	Bandwidth resources used by $h_k^{i_1,i_2}$.
Binary Decision Variables	
$z_{k,i}^n$	$z_{k,i}^n = 1$ if $f_{k,i}$ is placed on node $v_n \in V_s$ or $z_{k,i}^n = 0$.
$w_{i_1,i_2}^{k,p}$	$w_{i_1,i_2}^{k,p} = 1$ if the path p is used by $h_k^{i_1,i_2}$ or $w_{i_1,i_2}^{k,p} = 0$.
Variables	
x_n	$x_n = 1$ if server or core switch v_n is used or $x_n = 0$.
y_l	$y_l = 1$ if link l is used or $y_l = 0$.
e_l^p	$e_l^p = 1$ if link l is used by path p or $e_l^p = 0$.
U_{svr}	Utilization of servers in network.
U_L	Utilization of links in network.
U_B	Utilization of bandwidths in network.
U	Objective function.
∂	Weight value.

can be prior known. In order to improve the operating efficiency of satellite ground station networks, we investigate the problem of VNF placement and routing traffic by prior sensing resource requirements and running periods for user services.

For satellite communication systems, user service q_k is executed by satellites and the produced data should be sent back to the satellite ground station according to the satellite mission planning. Satellite ground station needs to provide the required resources for user service q_k in time and deploy the VNFs on available servers to further handle this data. The ingress s_k and egress d_k for user service q_k should be deployed on two different core switches. We place the adjacent VNFs from a user service on the same server as soon as possible to save the bandwidth resources. In addition, we should further improve the resource utilization of active servers to reduce the number of servers used by user services. Our objective is to minimize the number of used servers and the link resource utilization for satellite ground station networks. We assume that network resource allocation for all user services are handled in a batch processing mode. We collect the user services that are appearing in the next time slot and assign available network resources to them at a specific time interval. The resource allocation algorithm is implemented based on predicting resource requirements and running periods of user services according to the satellite mission planning.

4 Problem Formulation

4.1 Problem Description

In this section, we provide the problem description for VNF placement and routing traffic with mathematical methods. For satellite ground station networks, our goal is to maximize the resource utilization of active servers to save the energy cost. That is, the number of servers used by user services is as small as possible. Simultaneously, we expect to minimize the resource utilization of bandwidths and links. To address the problem of resource allocation, we formulate an ILP model. The main symbols used in our problem description are summarized in Table 1.

In order to better describe the problem of VNF placement and routing traffic, we denote a path between two servers or a server and a core switch as p. The variable P_{n_1,n_2} indicates the set of the shortest d paths between $v_{n_1} \in V_s$ and $v_{n_2} \in V_s$. The variable P is denoted as the set of all paths for each source and destination pair, which can be obtained in advance.

We denote a variable $x_n = \{0, 1\}$ to represent the active state of server or core switch v_n.

$$x_n = \begin{cases} 1 & \text{if server or core switch } v_n \text{ is used,} \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

A variable $y_l = \{0, 1\}$ indicates whether the l-th link is used or not.

$$y_l = \begin{cases} 1 & \text{if link } l \text{ is used,} \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

When two adjacent VNFs from a user service are deployed on two different servers, a path p between the two servers will be selected to route traffic flows. A variable e_l^p is used to represent whether link l is used by path p or not.

$$e_l^p = \begin{cases} 1 & \text{if link } l \text{ is used by path } p, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

We define a binary decision variable $z_{k,i}^n = \{0, 1\}$ to express whether VNF $f_{k,i}$ is placed on server or core switch v_n.

$$z_{k,i}^n = \begin{cases} 1 & \text{if VNF } f_{k,i} \text{ is placed on } v_n, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

We also define a binary decision variable $w_{i_1,i_2}^{k,p}$ to indicate which path p is used by the edge $h_k^{i_1,i_2}$. If path p offers the traffic flows for $h_k^{i_1,i_2}$, then $w_{i_1,i_2}^{k,p} = 1$, otherwise the value is 0.

For each VNF $f_{k,i} \in q_k$, it can be deployed on one and only one server or core switch $v_n \in V_s$. This constraint is represented as follows:

$$\sum_{v_n \in V_s} z_{k,i}^n = 1, \forall f_{k,i} \in q_k. \tag{5}$$

In our problem formulation, we assume that the ingress and egress of each service should be processed on two different core switches, respectively. So that we need to ensure that s_k and d_k for service q_k are placed on core switches. We express this constraint as follows:

$$z^n_{k,i} \cdot (1 - x_n) = 0, f_{k,i} = s_k, d_k, \forall v_n \in V_{cs}. \quad (6)$$

If two adjacent VNFs from a user service are allocated on two servers or a server and a core switch, then we need to ensure that a path p between the two network nodes can be provisioned. The constraint is described in equation (7) below.

$$z^{n_1}_{k,i_1} \cdot z^{n_2}_{k,i_2} = \sum_{p \in P_{n_1,n_2}} w^{k,p}_{i_1,i_2}, \forall v_{n_1}, v_{n_2} \in V_s, h^{i_1,i_2}_k \in H_k. \quad (7)$$

For a physical network, resource capacities of nodes and links are limited. The physical resource constraints should be guaranteed when we place VNFs to network nodes and route traffic flows. In this paper, we consider the resource requirements of CPU and Memory for servers.

We need to ensure that the total resource requirements for user services on a physical server can not exceed its resource capacity. The resource constraint for each network node is indicated as follows:

$$\sum_{q_k \in Q} \sum_{f_{k,i} \in q_k} z^n_{k,i} \cdot c^r_{k,i} \leq x_n \cdot C^r_n, \forall v_n \in V_s, r \in R. \quad (8)$$

We also need to ensure that the resource constraint for each physical link can be satisfied. The used bandwidths for a physical link should be less than its resource capacity. The related constraint is depicted as follows:

$$\sum_{q_k \in Q} \sum_{h^{i_1,i_2}_k \in H_k} \sum_{p \in P} w^{k,p}_{i_1,i_2} \cdot e^p_l \cdot b^{i_1,i_2}_k \leq y_l \cdot B_l, \forall l \in E. \quad (9)$$

In this paper, our objective is to minimize the resource utilization of the physical network, including servers, bandwidths and links.

The total number of active servers in the physical network is described as $\sum_{v_n \in V_{svr}} x_n$, then the utilization U_{svr} of servers can be represented as follows:

$$U_{svr} = \frac{1}{N_{svr}} \cdot \sum_{v_n \in V_{svr}} x_n. \quad (10)$$

The total number of active links is expressed as $\sum_{l \in E} y_l$, and the link utilization U_L is indicated as follows:

$$U_L = \frac{1}{L_e} \cdot \sum_{l \in E} y_l. \quad (11)$$

For service q_k, we denote the used bandwidth resources of link l as $U^l_{B,k}$ which can be expressed as:

$$U^l_{B,k} = \sum_{h^{i_1,i_2}_k \in H_k} \sum_{p \in P} w^{k,p}_{i_1,i_2} \cdot e^p_l \cdot b^{i_1,i_2}_k, \forall l \in E, q_k \in Q, \quad (12)$$

then the total bandwidth utilization $U^l_{B,Q}$ for link l can be described as follows:

$$U^l_{B,Q} = \frac{1}{B_l} \cdot \sum_{q_k \in Q} U^l_{B,k}, \forall l \in E. \quad (13)$$

Based on the above discussion, the total bandwidth utilization U_B in the physical network is represented as follows:

$$U_B = \frac{1}{L_e} \cdot \sum_{l \in E} U^l_{B,Q}. \quad (14)$$

Our objective function U can be expressed as a weighted sum of U_{svr}, U_L and U_B [7].

$$U = \partial_{svr} \cdot U_{svr} + \partial_L \cdot U_L + \partial_B \cdot U_B, \quad (15)$$

where ∂_{svr}, ∂_L and ∂_B are the weight factors, which can be used to adjust the preferences of different resources. We consider that $\partial_{svr} + \partial_L + \partial_B = 1$. The problem of VNF placement and routing traffic is formulated as ILP problem and the objective is to minimize the resource utilization of the underlying network with the physical resource constraints. It can be described as follows:

$$\begin{aligned} \min \quad & U \\ s.t. \quad & (1) - (9). \end{aligned} \quad (16)$$

In the next subsection we discuss the complexity analysis of the resource allocation problem.

4.2 Complexity Analysis

The problem of resource allocation in equation (16) can be seen as NP-hard due to the fact that a single source capacitated facility location problem (SSCFLP) [43] can be reduced to our formulated problem.

For SSCFLP, there are pre-specified sites J and customers I, respectively. The operating cost is denoted as f_i and the transportation cost for customer j is denoted as $c_{i,j}$ when a facility is located at a site i. The capacity of a facility at a site i is defined by s_i, and the demand of customer j is defined by w_j. A binary variable y_i indicates whether a facility is located at site i. A binary variable $x_{i,j}$ represents whether the demand of customer j is offered by a facility at site i. The problem of SSCFLP can be described as follows [43]:

$$\begin{aligned} \min \quad & \sum_{i \in I} \sum_{j \in J} c_{ij} \cdot x_{ij} + \sum_{i \in I} f_i \cdot y_i \\ s.t. \quad & \sum_{i \in I} x_{ij} = 1, \forall j \in J \\ & \sum_{j \in J} w_j \cdot x_{ij} \leq s_i \cdot y_i, \forall i \in I \\ & x_{ij} \in \{0,1\}, y_i \in \{0,1\}, \forall i \in I, j \in J. \end{aligned} \quad (17)$$

In order to reduce SSCFLP to the problem of VNF placement and routing traffic in this paper, we need to redescribe our optimization problem of resource allocation. Similar to reference [8], a user service is represented as $facility \rightarrow customer$, where all VNFs of q_k except d_k are regarded as a commodity to run in a facility and d_k is a customer. We set a server to be a facility and the resource capacity of a server is equal to the capacity of a facility. The resource demand of a user server on a server can be described as the demand of a customer in a facility. In addition, the resource utilization of a server represents the running cost for a facility. The used links and bandwidths for a user service can be indicated as the transportation cost from a facility to a customer. Further, we make a customer for service q_k locate on a core switch that is used by d_k, and

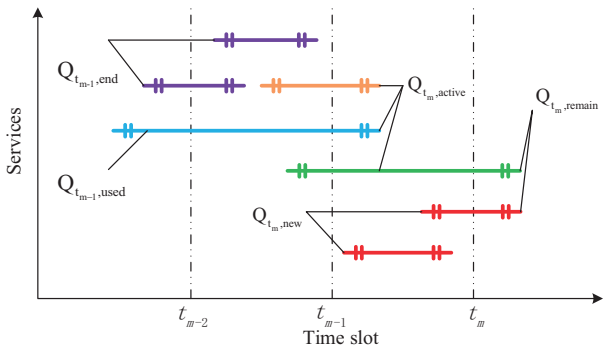


Fig. 4. Procedure for running LARA algorithm.

path p is used to route traffic flows. We ensure that the used bandwidth resources for each link are not limited. Then we can transform SSCFLP to the problem of VNF placement and routing traffic. SSCFLP is well-known as NP-hard, so the problem of resource allocation in this paper is also NP-hard.

5 Proposed Algorithms

As the problem of resource allocation is NP-hard, to optimize the resource utilization, we propose two location-aware resource allocation algorithms based on Greedy and CPLEX, respectively. Firstly, we implement the location-aware resource allocation algorithm by IBM CPLEX solver with version 12.10. However, with the increase in the number of user services and scale of network, the computational complexity of solving the NP-hard problem by CPLEX increases rapidly and we must take a long computational time for addressing the problem of resource allocation. So the proposed LARA algorithm based on CPLEX is not suitable to be used in large scale problem of resource allocation. In order to solve the VNF placement and routing traffic in the large scale problem, we also achieve the location-aware resource allocation algorithm based on Greedy to obtain an approximate solution.

5.1 Location-Aware Resource Allocation Algorithm

For satellite ground station networks, we can know the information prior about service type, resource requirements and life cycle time for all user services depending upon the satellite mission planning in satellite control center. In view of predictable user services, we propose the location-aware resource allocation algorithm to effectively reduce the network resource utilization in terms of servers, bandwidths and links.

The procedure of resource allocation in a time slot is divided into two parts as: (1) finding an optimization solution and (2) VNF placement and routing traffic. At the beginning of a time slot, the proposed LARA algorithm is used to seek an optimization solution of resource allocation. As the results of the optimization solution, we can deploy the VNFs and select routing traffics for current requested user services. The total time of the two procedures should be less than a time slot interval. For our proposed LARA algorithm, when we look for the optimization solution of resource allocation, we can predict the resource requirement

Algorithm 1 Location-Aware Resource Allocation Algorithm.

Input: Time slot t, number of predictable time slots M;
Output: Feasible solution;

1: **Initialize:** $m = M$,$Q_{t_m,remain} = null$;
2: **while** $m > 0$ **do**
3: $t_m \leftarrow t + m$;
4: Obtain new user services $Q_{t_m,new}$ to be allocated resources in time slot t_m;
5: Find all active services $Q_{t_m,active}$ at the beginning of time slot t_m;
6: Get active services $Q_{t_{m-1},used}$ that are offered resources before time slot t_m;
7: Acquire services $Q_{t_{m-1},end}$ that are finished before time slot t_m;
8: $Q_{t_m-1,remain} \leftarrow Q_{t_m,active} - Q_{t_{m-1},used}$;
9: $Q_{t_m,allocate} \leftarrow \{Q_{t_m-1,remain}, Q_{t_m,new}\} - Q_{t_m,remain}$;
10: Free server and bandwidth resources used by $Q_{t_{m-1},end}$;
11: Allocate the resources of servers and links for $Q_{t_m,remain}$;
12: Search an optimization solution of resource allocation for $Q_{t_m,allocate}$ by $Greedy$ or $CPLEX$;
13: $m \leftarrow m - 1$;
14: $Q_{t_m,remain} \leftarrow \{Q_{t_m-1,remain}, Q_{t_m,remain}\} - Q_{t_m,new}$;
15: **end while**
16: **return** Optimization solution for $Q_{t,new}$;

and running state information about user services in the future multiple time slots according to the satellite mission planning. Our purpose of resource allocation is to minimize the resource utilization in the predictable time slots as soon as possible.

Fig. 4 shows the procedure for running our proposed LARA algorithm in predictable time slot t_m. We denote current time slot as t and the predictable time slot as t_m. All active user services are classified into five types according to their running states in different time slots and described as follows:

- *Service-type1:* For predictable time slot t_m, if user services in active states are over before time slot t_m, we can indicate them by $Q_{t_{m-1},end}$ and the user services from $Q_{t_{m-1},end}$ are considered as service-type1.
- *Service-type2:* At the beginning of time slot t_m, the user services that are still active are considered as service-type2 and denoted by $Q_{t_m,active}$.
- *Service-type3:* The user services that are assigned network resources before time slot t_m are considered as service-type3 and represented by $Q_{t_{m-1},used}$.
- *Service-type4:* $Q_{t_m,new}$ indicates the user services that are occuring in time slot t_m. Let us denote user services in $Q_{t_m,new}$ as service-type4.
- *Service-type5:* $Q_{t_m,remain}$ expresses the user services that are allocated network resources during $[t, t_m]$ time slots and also active in time slot t_{m+1}. Let $Q_{t_m,remain}$ be service-type5.

Based on the above discussion, $Q_{t_m-1,remain}$ can be ob-

Algorithm 2 Greedy Algorithm.

Input: User services $Q_{t_m,allocate}$;
Output: Feasible solution;

Collect active servers $V_{svr,active}$ and idle servers $V_{svr,idle}$;
for each $q_k \in Q_{t_m,allocate}$ **do**
  $flag, server \leftarrow Search(q_k, V_{svr,active})$;
  if $flag = false$ **then**
    $flag, server \leftarrow Search(q_k, V_{svr,idle})$;
    Add $server$ to $V_{svr,active}$;
    Remove $server$ from $V_{svr,idle}$;
  end if
end for
return Feasible solution for $Q_{t_m,allocate}$;

tained by:

$$Q_{t_m-1,remain} = Q_{t_m,active} - Q_{t_m,used}, \tag{18}$$

then we can obtain the user services $Q_{t_m,allocate}$ that need to be assigned in time slot t_m as follows:

$$Q_{t_m,allocate} = \{Q_{t_m-1,remain}, Q_{t_m,new}\} - Q_{t_m,remain}. \tag{19}$$

To effectively improve the resource utilization, we free the network resources used by user services in $Q_{t_m,end}$ and deploy the available network resources to the user services in $Q_{t_m,remain}$ by the results of resource allocation that were computed in time slot t_{m+1}. Then the Greedy and CPLEX approaches are carried out to find an optimization solution of resource allocation for the user services in $Q_{t_m,allocate}$. After that, $Q_{t_m,remain}$ can be updated by:

$$Q_{t_m,remain} = \{Q_{t_m-1,remain}, Q_{t_m,remain}\} - Q_{t_m,new}. \tag{20}$$

The procedure of our proposed LARA algorithm is described in Algorithm 1. Current time slot is t and the number of predicted time slots is M. At the beginning, we set $m = M$ and $Q_{t_m,remain} = null$. For time slot t_m, firstly we can predict $Q_{t_{m-1},end}$, $Q_{t_m,remain}$ and $Q_{t_m,allocate}$, respectively. Then we free the network resources used by user services in $Q_{t_{m-1},end}$, and allocate resources to user servers in $Q_{t_m,remain}$. Greedy and CPLEX algorithms are executed to find an optimization solution of resource allocation for user services in $Q_{t_m,allocate}$. The procedure of our proposed LARA algorithm can be executed M times and then we can obtain an optimization solution of resource allocation for $Q_{t,new}$.

For the proposed LARA algorithm based on CPLEX, we address the ILP problem of resource allocation by IBM CPLEX solver tool with version 12.10, which is configured by default algorithm parameters and can obtain a global optimization solution of resource allocation.

In the following subsection we discuss the Greedy algorithm used by our proposed LARA algorithm.

5.2 Greedy Algorithm

In this paper, our proposed LARA algorithm is implemented by Greedy to address the problem of resource allocation. The processing of Greedy algorithm is shown in Algorithm 2. The input parameters are user services

Algorithm 3 Search.

Input: User service q_k, collection of servers $\tilde{V}_{svr}$;
Output: $success, server$;

$success = false, server = null$;
for each $v_n \in \tilde{V}_{svr}$ **do**
  Obtain the VNF sequence Γ_k of q_k using topological sort method;
  for each $f_{k,i} \in \Gamma_k$ **do**
    if $f_{k,i} \notin [s_k, d_k]$ **then**
      Attempt to place function $f_{k,i}$ to sever v_n;
      if v_n is not provide enough resources to $f_{k,i}$ **then**
        Break;
      end if
    else
      v_n is updated as a core switch used by s_k or d_k;
    end if
    Get all predecessors of $f_{k,i}$ and their edges $H^{pre}_{k,i}$;
    for each $h^{\tilde{i},i}_k \in H^{pre}_{k,i}$ **do**
      Find server $v_{\tilde{n}}$ used by $f_{k,\tilde{i}}$;
      Sort $p_{\tilde{n},n}$ between $v_{\tilde{n}}$ and v_n by the path distance;
      for each $p \in p_{\tilde{n},n}$ **do**
        Calculate available bandwidths for $h^{\tilde{i},i}_k$;
        if there are enough bandwidths for $h^{\tilde{i},i}_k$ **then**
          Break;
        end if
      end for
    end for
  end for
  if q_k can allocate to v_n **then**
    Perform objective function U;
    if Objective value is better than others **then**
      $server = v_n$;
    end if
    $success = true$;
  end if
end for
return $success, server$;

$Q_{t_m,allocate}$. At the beginning, we divide all available servers in the physical network into two portions. One is that the servers used by user services are indicated as $V_{svr,active}$. the other is that the servers in idle states are indicated as $V_{svr,idle}$. For user service $q_k \in Q_{t_m,allocate}$, firstly we call function $Search$ to seek a feasible solution from servers in $V_{svr,active}$ to minimize the resource utilization. If any server in $V_{svr,active}$ can not be used by q_k, then $flag = false$, or otherwise $flag = true$. When $flag = false$ we will find a feasible solution from servers in $V_{svr,idle}$ by function $Search$. If a server in $V_{svr,idle}$ is selected to deploy user service q_k, the server should be moved from $V_{svr,idle}$ to $V_{svr,active}$ and it will be in active state. When all user services in $Q_{t_m,allocate}$ are assigned to the physical network, the Greedy algorithm will return a feasible solution. Note that we assume that satellite ground station network can provide enough resources for all user services.

Function $Search$ is designed to deploy the VNFs on

TABLE 2
Parameter Settings for Performance Evaluation

Network architectures					
Topology	Fat-Tree	BCube			VL2
Number of Servers	16	4, 8, 16			16
Resource capacities for servers					
Name	vCPUs			Memory	
Capacity	112			192 GB	
Resource capacities for links					
Name	link between a server and a switch			link between switches	
Capacity	1 Gbps			10 Gbps	
Configurations for user services					
Boundary	VNFs	Out-degree	vCPUs	Memory	Bandwidth
Lower	6	1	4	8 GB	100 Mbps
Upper	10	2	8	12 GB	200 Mbps

servers, and select routing traffic for the edge between two adjacent VNFs on different nodes. The aim is to minimize the resource utilization of servers, links and bandwidths. The input parameters include user service q_k and a set $\tilde{V}_{svr}$ of servers. The output parameters are an identification "$success$" of success and a server "$server$" used by q_k.

Initially, we set $success = false$ and $server = null$. For each server $v_n \in \tilde{V}_{svr}$, we attempt to deploy q_k to server v_n. Firstly, the sequence Γ_k of VNFs for q_k is obtained by a topology sort method to ensure that source f_{k,i_1} comes before sink f_{k,i_2} for edge (f_{k,i_1}, f_{k,i_2}). For each VNF $f_{k,i} \in \Gamma_k$, we place VNF $f_{k,i}$ to server v_n. If server v_n can not provide the resource demands of $f_{k,i}$, then we will break the loop and turn to the next server to deploy q_k, otherwise we will obtain all predecessors of $f_{k,i}$ and the edges $H^{pre}_{k,i}$ between $f_{k,i}$ and its predecessors. For each edge $h^{\tilde{i},i}_k \in H^{pre}_{k,i}$, we search the host server $v_{\tilde{n}}$ for $f_{k,\tilde{i}}$, and sort all paths in $p_{\tilde{n},n}$ by the path distance. Then we calculate available bandwidths of each path $p \in p_{\tilde{n},n}$ for edge $h^{\tilde{i},i}_k$. If the bandwidth demands of edge $h^{\tilde{i},i}_k$ are not offered by any path $p \in p_{\tilde{n},n}$, the loop is also broken. When service q_k can be deployed to server v_n, the objective function will be performed. If the value for server v_n is smaller than that of others, then $server = v_n$ and $success = true$. Function $Search$ is described in Algorithm 3.

6 Performance Evaluation

In this section, we make the experiments to evaluate the performance of the proposed LARA algorithms based on Greedy and IBM CPLEX 12.10, respectively. In small scale networks, we discuss the solution quality and computational complexity of our proposed Greedy- and CPLEX-based LARA algorithms in addressing the problem of VNF placement and routing traffic. Furthermore, we evaluate the performance of the proposed Greedy-based LARA algorithm for different predictable time slots in large scale networks. The experimental platform is a commodity server, which includes i7-4790K CPU, 16 GB of RAM and windows 10. PYTHON is used as programming language.

6.1 Simulation Setup

In our performance evaluation, the weight values in equation (15) are set as $\partial_{svr} = \partial_L = \partial_B = \frac{1}{3}$. The time slot interval is 10 minutes. Similar to reference [7], three network structures of Fat-Tree, BCube and VL2 are considered to run our experiments.

- *Fat-Tree:* Fat-Tree [26] is a layered-structure network with core layer, aggregation layer and top-of-rack layer, and can be widely used in data center networks. A k fat-tree network indicates that there are k ports for each switch. It consists of $(\frac{k}{2})^2$ core switches and k pods, where each pod include k switches [7].
- *BCube:* BCube is a server-centric network structure for shipping-container based modular data centers. Each server has several switch ports and can connect to multiple switches of different levels. For $BCube_0$, n servers connect to a switch with n ports. A $BCube_k (k \geq 1)$ is constructed by n $BCube_{k-1}s$ and n^k switches with n ports. There are n^{k+1} servers and $k+1$ levels of switches for $BCube_k$ [27].
- *VL2:* VL2 is a scalable and flexible network to support large data centers that are uniform high capacity between servers and can achieve performance isolation between services. It is composed of server layer and switch layer. Servers are connected to the switch layer by top-of-rack switches. A complete bipartite graph is formed by the links between aggregation and intermediate switches [28]. For k-port aggregation switches and n-port top-of-rack switches, VL2 consists of $n \cdot \frac{k^2}{4}$ servers.

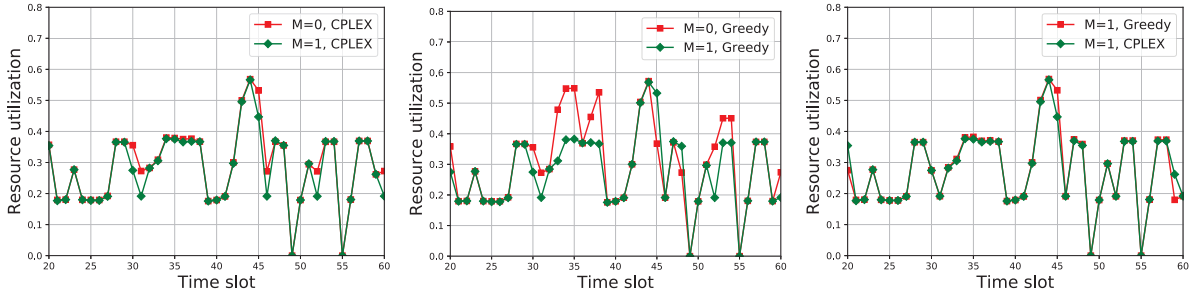
Due to the computational complexity of solving an ILP problem by CPLEX, the effectiveness of our proposed LARA algorithms based on Greedy and CPLEX is demonstrated in small data center network $BCube$ with 4 and 8 servers. Then we evaluate the performance of the proposed Greedy-based LARA algorithm for various predictable time slots in three networks of Fat-Tree, BCube and VL2, where the number of servers is 16. We assume that all servers have the same resource configurations. The resource capacities for each server are 112 vCPUs and 192 GB Memory. We set the bandwidth capacity for each link between a server and a switch as 1 Gbps and for each link between switches as 10 Gbps [28]. The shortest $d = 8$ paths between a core switch and a server or two servers can be computed in advance.

To simplify our simulation experiments, we assume that all user services are provided by Low Earth Orbit (LEO) satellites and each user service can observe a fixed objective on the ground periodically. The life cycle time for all user services can be obtained by the Satellite Tool Kit (STK) and prior known for satellite ground station networks. We randomly generate the VNFs and their resource demands. The number of VNFs for a user service is ranged from 6 to 10. The maximum number of outgoing edges for each VNF can be 2. The resource requirements of a VNF are $[4, 8]$ units for vCPUs and $[8, 12]$ GB for Memory. The bandwidth demands between two adjacent VNFs is randomly gained from $[100, 200]$ Mbps.

The main parameter settings used in the performance evaluation are listed in Table 2.

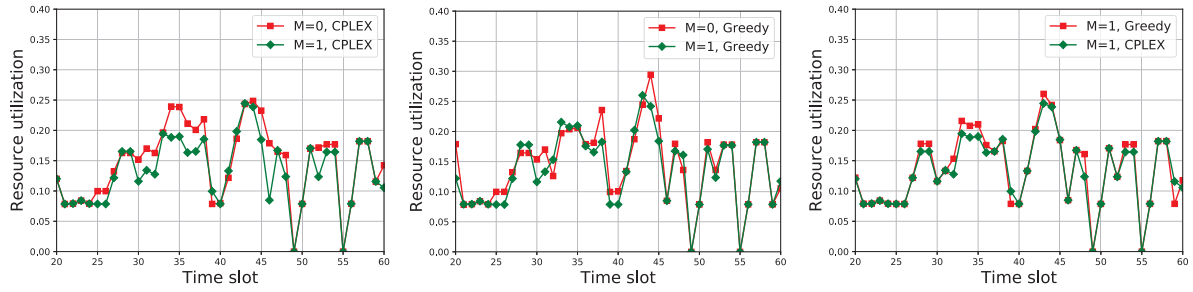
6.2 Performance Comparison of Greedy and CPLEX

In this section, we simulate and evaluate the performance of our proposed LARA algorithms based on Greedy and



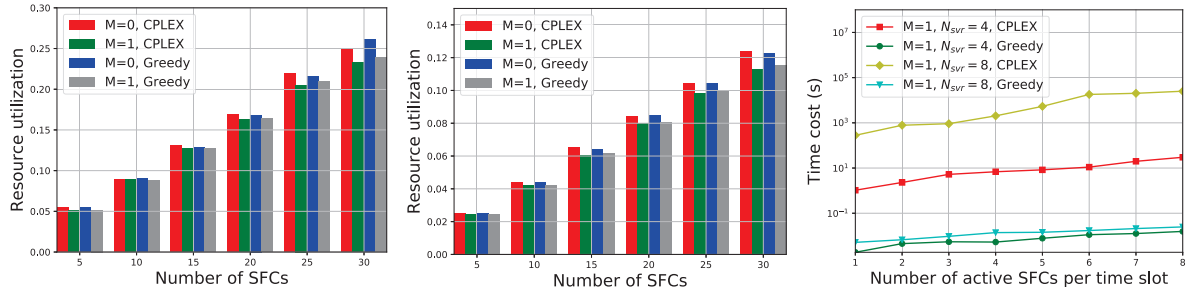
(a) LARA algorithm by CPLEX (b) LARA algorithm by Greedy (c) Results of LARA algorithms

Fig. 5. Results of LARA algorithms in BCube network with 4 servers.



(a) LARA algorithm by CPLEX (b) LARA algorithm by Greedy (c) Results of LARA algorithms

Fig. 6. Results of LARA algorithms in BCube network with 8 servers.



(a) BCube with 4 servers (b) BCube with 8 servers (c) Time cost

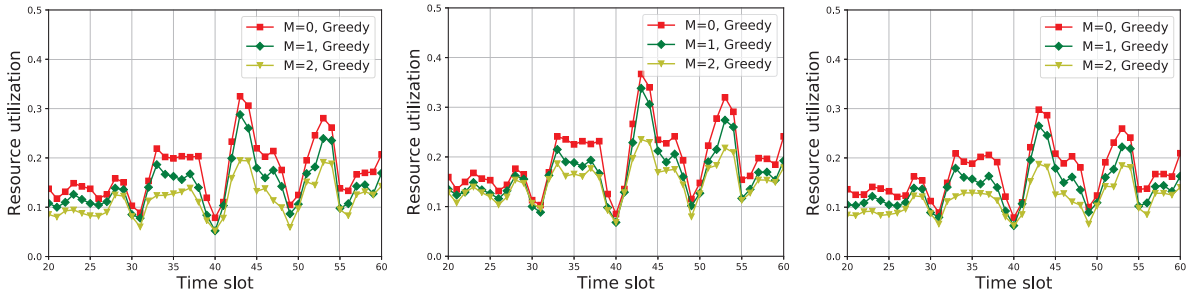
Fig. 7. Performance comparison between Greedy and CPLEX in BCube network.

CPLEX in small BCube networks, where the number of servers is 4 and 8, respectively. Two situations of predictable and un-predictable user services are taken into consideration in our experiments. Then we discuss the effectiveness of the two proposed LARA algorithms in terms of solution quality and computational cost.

Fig. 5 shows the results of our proposed LARA algorithm in BCube with 4 servers. The number of user services is set as 30. M indicates the number of predictable time slots, $M = 0$ means that the proposed LARA algorithm can not predict the life cycle time of user services. Fig. 5(a) and Fig. 5(b) describe the total resource utilizations of BCube obtained by the proposed LARA algorithms based on Greedy and CPLEX, respectively. From Fig. 5(a) and Fig. 5(b), it is obvious that the proposed LARA algorithms with the predictable functionality performs better than the conventional resource allocation algorithms. In Fig. 5(c), we show the resource utilization results of the proposed LARA algorithms with one predictable time slot. We can observe that the proposed LARA algorithms achieved by Greedy and CPLEX have very similar performance.

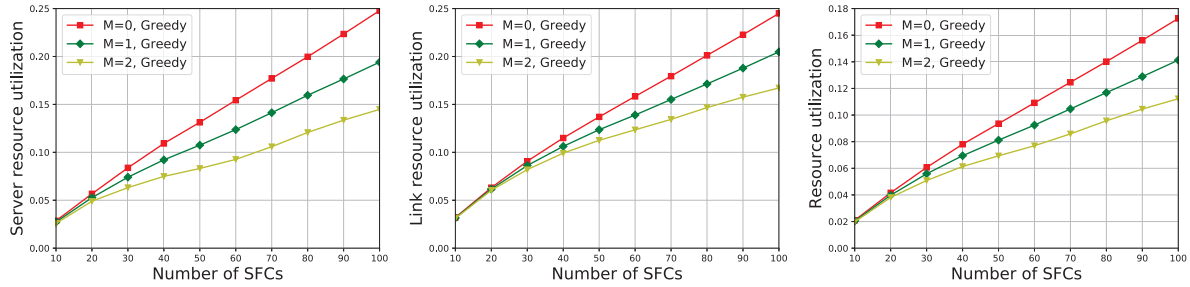
Similar results are shown for BCube network with 8 servers in Fig. 6. Fig. 6(a) and Fig. 6(b) describe the results of our proposed LARA algorithms based on Greedy and CPLEX, respectively. The performance comparison of our proposed LARA algorithms based on Greedy and CPLEX is illustrated in Fig. 6(c). Compared with the results as shown in Fig. 5, the performance gap between the proposed LARA algorithm and the conventional resource allocation algorithm could be more obvious in BCube network with 8 servers. However, we can observe that our proposed LARA algorithm is an effective approach to improve the performance of solving the problem of VNF placement and routing traffic according to prior sensing the running conditions of satellite user services.

In addition, our experiments for different number of user services are carried out to evaluate the performance of the proposed LARA algorithm. The average resource



(a) Fat-Tree network (b) BCube network (c) LV2 network

Fig. 8. Resource utilizations for Fat-Tree, BCube and LV2.



(a) Resource utilization for servers (b) Resource utilization for links (c) Resource utilization

Fig. 9. Resource utilizations for Fat-Tree with 16 servers.

utilizations per time slot for various number of user services are shown in Fig. 7. The number of user services is denoted as $[5, 10, 15, 20, 25, 30]$ and the running time for each user service is 24 hours. The results of average resource utilizations in BCube networks with 4 and 8 servers are depicted in Fig. 7(a) and Fig. 7(b), respectively. In all cases of resource allocation, we can find from Fig. 7(a) and Fig. 7(b) that the performance of our proposed LARA algorithm is better than that of the conventional resource allocation algorithm. Furthermore, the two proposed LARA algorithms based on Greedy and CPLEX show close results in seeking the solution of resource allocation. For example, the resource utilizations obtained by the proposed LARA algorithms based on Greedy and CPLEX are 0.2332 and 0.2391 for $N_{svr} = 4, K = 30, M = 1$, and 0.1132 and 0.1151 for $N_{svr} = 8, K = 30, M = 1$, respectively.

The computational time costs for the proposed LARA algorithms based on CPLEX and greedy are described in Fig. 7(c). Here we consider that the number of active user services per time slot is $[1, \ldots, 8]$ due to the computational complexity of CPLEX. BCube networks consist of 4 and 8 servers, respectively. We can find that the proposed LARA algorithm based on CPLEX has a long running time for addressing the problem of VNF placement and routing traffic, especially, with the increase in scale of network and number of active user services. However, our proposed Greedy-based LARA algorithm can quickly obtain an approximated solution for solving the problem of resource allocation. In BCube with 4 servers, when there are 4 active user services, the average time cost is 6.8507 seconds for CPLEX and 0.0053 seconds for Greedy. When the number of active user services is 9, the average time cost can be 29.9021 seconds for CPLEX and 0.0156 seconds for Greedy. In BCube with 8 servers, when there are 2 active user services, the average time cost can be 13.0439 minutes for CPLEX and 0.0067 seconds for Greedy. When the number of active user services is 5, the average time cost can be 89.3250 minutes for CPLEX and 0.0143 seconds for Greedy. We can find that the proposed LARA algorithm based on CPLEX can address the problem of VNF placement and routing traffic in small scale networks, however, it is not suitable to be used in large scale networks. The proposed LARA algorithm based on Greedy in this paper is an effective approach of resource allocation to address the problem of VNF placement and routing traffic in large scale networks.

6.3 Performance Analysis of Greedy-based LARA Algorithm

In this section, we evaluate the performance of the proposed Greedy-based LARA algorithm for multiple predictable time slots in three network structures of Fat-Tree, BCube and LV2 with 16 servers, respectively. The number of user services is from 10 to 100 and the durations for all user services are 7×24 hours. The predictable time slots are $0, 1$ and 2, respectively. Each experiment is carried out 30 times and we obtain the average results of resource utilizations in terms of servers, bandwidths and links.

Fig. 8 shows the results of resource utilizations obtained by the proposed Greedy-based LARA algorithm for 90 user services in Fat-Tree, BCube and LV2 networks, respectively. In Fig. 8(a), the results of resource utilizations for $M = 0, 1$ and 2 in Fat-Tree are illustrated. We can observe that the performance of the proposed Greedy-based LARA algorithm becomes better as the number of predictable time slots increases for our simulation parameters setup. The proposed Greedy-based LARA algorithm performs better

TABLE 3
Resource utilizations for Fat-Tree, BCube and LV2

K	Fat-Tree			BCube			LV2		
	M=0	M=1	M=2	M=0	M=1	M=2	M=0	M=1	M=2
10	0.0210	0.0204	0.0197	0.0244	0.0238	0.0238	0.0249	0.0243	0.0235
20	0.0416	0.0398	0.0381	0.0497	0.0478	0.0479	0.0486	0.0466	0.0447
30	0.0606	0.0559	0.0508	0.0717	0.0675	0.0669	0.0688	0.0638	0.0583
40	0.0780	0.0694	0.0613	0.0913	0.0848	0.0830	0.0851	0.0768	0.0682
50	0.0936	0.0812	0.0694	0.1086	0.0996	0.0964	0.0999	0.0884	0.0760
60	0.1092	0.0925	0.0770	0.1269	0.1137	0.1084	0.1146	0.0990	0.0833
70	0.1246	0.1046	0.0859	0.1429	0.1259	0.1192	0.1269	0.1080	0.0903
80	0.1402	0.1169	0.0957	0.1610	0.1389	0.1285	0.1412	0.1194	0.0991
90	0.1562	0.1289	0.1045	0.1762	0.1514	0.1384	0.1535	0.1286	0.1060
100	0.1726	0.1413	0.1123	0.1940	0.1653	0.1479	0.1704	0.1407	0.1142

than the conventional Greedy-based resource allocation algorithm. The proposed Greedy-based LARA algorithm for $M = 2$ outperforms that of $M = 1$. Similar results for BCube and LV2 networks can be found in Fig. 8(b) and Fig. 8(c), respectively. It is obvious that our proposed Greedy-based LARA algorithm can effectively decrease the resource utilization of the three networks by introducing the predictable functionality.

To further investigate the influence of different number of user services on the performance, we run the experiments for $K = [10, 20, \cdots, 100]$ by the proposed Greedy-based LARA algorithm in Fat-Tree network with 16 servers and the average results of resource utilizations are shown in Fig. 9. The proposed Greedy-based LARA algorithm with $M = 0$ is considered as our baseline algorithm. Fig. 9(a) illustrates the resource utilization of servers for different number of user services. We can observe that the proposed Greedy-based LARA algorithms for $M = 0, 1$ and 2 have relatively close results in the case of small number of user services, and our proposed Greedy-based LARA algorithm performs better with the increase in the number of user services and predictable time slots, respectively. For instance, in the case of $K = 50$, the performance improvement of our proposed Greedy-based LARA algorithm in the resource utilization of servers is 18.13% for $M = 1$ and 36.66% for $M = 2$. On average, the resource utilization of servers obtained by the proposed Greedy-based LARA algorithm is 18.64% for $M = 1$ and 36.72% for $M = 2$ less than that of $M = 0$. The resource utilizations of links for different number of user services are shown in Fig. 9(b). We can observe that our proposed Greedy-based LARA algorithm effectively decreases the number of used links in assigning network resources for user services. For $K = 50$, the resource utilization of links obtained by our proposed Greedy-based LARA algorithm reduces by 9.78% for $M = 1$ and 17.87% for $M = 2$. On average, the link resource utilization of our proposed Greedy-based LARA algorithm saves by 12.26% for $M = 1$ and 22.85% for $M = 2$. The total resource utilizations for different user services are described in Fig. 9(c). As shown in Fig. 9(c), the average resource utilization gained by our proposed Greedy-based LARA algorithm decreases by 14.71% for $M = 1$ and 28.35% for $M = 2$, respectively.

In order to evaluate the performance of our proposed Greedy-based LARA algorithm in three network structures of Fat-Tree, BCube and LV2, we make the experiments for different user services in Fat-Tree, BCube and LV2. Each experiment is carried out 30 times and the average results of resource utilizations are shown in Table. 3. We can observe that our proposed Greedy-based LARA algorithm performs better than the Greedy-based resource allocation algorithm in these three networks. For $M = 1$, the average resource utilizations obtained by our proposed Greedy-based LARA algorithm for Fat-Tree, BCube and LV2 decrease by $14.71\%, 11.15\%$ and 13.38%, respectively. In the case of $M = 2$, our proposed Greedy-based LARA algorithm for Fat-Tree, BCube and LV2 has $28.5\%, 16.23\%$ and 26.14% performance improvement on average, respectively. Hence it can be stated that our proposed Greedy-based LARA algorithm can effectively improve the performance of resource utilizations for three networks of Fat-Tree, BCube and LV2.

7 Conclusion

In this paper, considering that the information about service type, resource requirements and life cycle time for user services can be known beforehand depending on the satellite mission planning in satellite control center, we investigate the optimization problem of location-aware resource allocation for satellite ground station networks. We formulate the problem of VNF placement and routing traffic as an integer linear programming model and prove it as NP-hard. Our goal is to minimize the resource utilization of the underlying network within the physical resource constraints.

To address this problem, The LARA algorithms based on Greedy and CPLEX are implemented. We simulate and evaluate the performance of the two proposed LARA algorithms in small scale networks of BCube with 4 and 8 servers, respectively. The results show that the proposed LARA algorithms based on CPLEX and Greedy have close performance, where the CPELX-based LARA algorithm can be used in small scale networks due to the computational complexity. To further discuss the performance of our proposed

LARA algorithm, we use the proposed Greedy-based LARA algorithm to address the problem of resource allocation in three networks of Fat-Tree, BCube and LV2 with 16 servers, respectively. We can find that our proposed Greedy-based LARA algorithm outperforms the Greedy-based resource allocation algorithm for the three networks in the resource utilizations of servers, links and total network resources. In addition, the number of predictable time slots has an effect on the performance of our proposed LARA algorithm. The resource utilizations of Fat-Tree, BCube and LV2 obtained by our proposed Greedy-based LARA algorithm can decrease by $14.71\%, 11.15\%$ and 13.38% for $M = 1$, and $28.5\%, 16.23\%$ and 26.14% for $M = 2$ on average.

REFERENCES

[1] J. H. Cox, J. Chung, S. Donovan *et al.*, "Advancing software-defined networks: A survey," *IEEE Access*, vol. 5, pp. 25 487–25 526, 2017.

[2] R. Mijumbi, J. Serrat, J. Gorricho *et al.*, "Network function virtualization: State-of-the-art and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 236–262, 2016.

[3] A. M. Medhat, T. Taleb, A. Elmangoush *et al.*, "Service function chaining in next generation networks: State of the art and research challenges," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 216–223, 2017.

[4] B. Kar, E. H. Wu, and Y. Lin, "Energy cost optimization in dynamic placement of virtualized network function chains," *IEEE Trans. Netw. Serv. Manag.*, vol. 15, no. 1, pp. 372–386, 2018.

[5] J. G. Herrera and J. F. Botero, "Resource allocation in NFV: A comprehensive survey," *IEEE Trans. Netw. Serv. Manag.*, vol. 13, no. 3, pp. 518–532, 2016.

[6] D. Bhamare, R. Jain, M. Samaka *et al.*, "A survey on service function chaining," *J. Netw. Comput. Appl.*, vol. 75, pp. 138–155, 2016.

[7] W. Rankothge, F. Le, A. Russo *et al.*, "Optimizing resource allocation for virtualized network functions in a cloud center using genetic algorithms," *IEEE Trans. Netw. Serv. Manag.*, vol. 14, no. 2, pp. 343–356, 2017.

[8] F. Bari, S. R. Chowdhury, R. Ahmed *et al.*, "Orchestrating virtualized network functions," *IEEE Trans. Netw. Serv. Manag.*, vol. 13, no. 4, pp. 725–739, 2016.

[9] F. Z. Yousaf, M. Bredel, S. Schaller *et al.*, "NFV and SDN-key technology enablers for 5G networks," *IEEE J. Sel. Area. Comm.*, vol. 35, no. 11, pp. 2468–2478, 2017.

[10] B. Feng, G. Li, G. Li *et al.*, "Enabling efficient service function chains at terrestrial-satellite hybrid cloud networks," *IEEE Netw.*, vol. 33, no. 6, pp. 94–99, 2019.

[11] T. Ahmed, R. Ferrus, R. Fedrizzi *et al.*, "Towards SDN/NFV-enabled satellite ground segment systems: Bandwidth on demand use case," in *Proc. IEEE Int Conf. Commun. Workshops*, Paris, France, Jun. 2017, pp. 894–899.

[12] F. Riffel and R. Gould, "Satellite ground station virtualization: Secure sharing of ground stations using software defined networking," in *Proc. Annu. IEEE Syst. Conf.*, Orlando,USA, Apr. 2016, pp. 1–8.

[13] Z. Wang, J. Zhang, T. Huang *et al.*, "Service function chain composition, placement, and assignment in data centers," *IEEE Trans. Netw. Serv. Manag.*, vol. 16, no. 4, pp. 1638–1650, 2019.

[14] Z. Li and Y. Yang, "Placement of virtual network functions in hybrid data center networks," *IEEE Trans. Multi-Scale Comput. Syst.*, vol. 4, no. 4, pp. 861–873, 2018.

[15] D. Li, P. Hong, K. Xue, and j. Pei, "Virtual network function placement considering resource optimization and SFC requests in cloud datacenter," *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 7, pp. 1664–1677, 2018.

[16] F. Xhafa, X. Herrero, A. Barolli *et al.*, "A tabu search algorithm for ground station scheduling problem," in *Proc. IEEE Int. Conf. Adv. Informa. Netw. Appl.*, Victoria, Canada, May 2014, pp. 1033–1040.

[17] Z. Xu, B. Lou, and C. Wang, "Task scheduling of satellite ground station systems based on the neighbor-area search algorithm," in *Proc. Int. Conf. Nat. Comput.*, Shenyang, China, Jul. 2013, pp. 1830–1834.

[18] A. Tepe and G. Yilmaz, "A survey on cloud computing technology and its application to satellite ground systems," in *Proc. Int. Conf. Recent Adv. Space Technol.*, Istanbul, Turkey, Jun. 2013, pp. 477–481.

[19] C. Fan, X. Zhao, L. Xie *et al.*, "A resource mapping method in cloud-based satellite ground system," in *Proc. IEEE Int. Conf. SmartCity/SocialCom/SustainCom*, Chengdu, China, Dec. 2015, pp. 1163–1166.

[20] D. Zhou, M. Sheng, X. Wang *et al.*, "Mission aware contact plan design in resource-limited small satellite networks," *IEEE Trans. Commun.*, vol. 65, no. 6, pp. 2451–2466, 2017.

[21] M. Tipaldi and L. Glielmo, "A survey on model-based mission planning and execution for autonomous spacecraft," *IEEE Syst. J.*, vol. 12, no. 4, pp. 3893–3905, 2018.

[22] F. Perea, R. Vazquez, and J. Galan-Viogue, "Swath-acquisition planning in multiple-satellite missions: an exact and heuristic approach," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 51, no. 3, pp. 1717–1725, 2015.

[23] E. Maurer, F. Mrowka, A. Braun *et al.*, "TerraSAR-X mission planning system: Automated command generation for spacecraft operations," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 2, pp. 642–648, 2010.

[24] J. Liu, Y. Li, Y. Zhang *et al.*, "Improve service chaining performance with optimized middlebox placement," *IEEE Trans. Serv. Comput.*, vol. 10, no. 4, pp. 560–573, 2015.

[25] IBM ILOG CPLEX optimization studio v12.10.0. [Online]. Available: https://www.ibm.com/support/knowledgecenter/SSSA5P_12.10.0/COS_KC_

[26] C. E. Leiserson, "Fat-trees: universal networks for hardware-efficient supercomputing," *IEEE trans. Comput.*, vol. 100, no. 10, pp. 892–901, 1985.

[27] C. Guo, G. Lu, D. Li *et al.*, "Bcube: a high performance, server-centric network architecture for modular data centers," in *Proc. ACM SIGCOMM*, Barcelona, Spain, Aug. 2009, pp. 63–74.

[28] A. Greenberg, J. R. Hamilton, N. Jain *et al.*, "VL2: a scalable and flexible data center network," in *Proc. ACM SIGCOMM*, Barcelona, Spain, Aug. 2009, pp. 51–62.

[29] M. A. Raayatpanah and T. Weise, "Virtual network function placement for service function chaining with minimum energy consumption," in *Pro. IEEE Int. Conf. Comput. Commun. Eng. Technol.*, Beijing, China, Aug. 2018, pp. 198–202.

[30] M. M. Tajiki, S. Salsano, L. Chiaraviglio *et al.*, "Joint energy efficient and QoS-aware path allocation and vnf placement for service function chaining," *IEEE Trans. Netw. Serv. Manag.*, vol. 16, no. 1, pp. 374–388, 2019.

[31] C. Assi, S. Ayoubi, N. El Khoury *et al.*, "Energy-aware mapping and scheduling of network flows with deadlines on VNFs," *IEEE Trans. Green Commun. Netw.*, vol. 3, no. 1, pp. 192–204, 2019.

[32] L. Qu, C. Assi, and K. Shaban, "Delay-aware scheduling and resource optimization with network function virtualization," *IEEE Trans. Commun.*, vol. 64, no. 9, pp. 3746–3758, 2016.

[33] B. Tan, H. Ma, and Y. Mei, "A NSGA-II-based approach for service resource allocation in cloud," in *Proc. IEEE Congr. Evol. Comput.*, San Sebastian, Spain, Jun. 2017, pp. 2574–2581.

[34] Q. Sun, P. Lu, W. Lu *et al.*, "Forecast-assisted NFV service chain deployment based on affiliation-aware vNF placement," in *Proc. GLOBECOM*, Washington, USA, Dec. 2016, pp. 1–6.

[35] H. Tang, D. Zhou, and D. Chen, "Dynamic network function instance scaling based on traffic forecasting and VNF placement in operator data centers," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 3, pp. 530–543, 2018.

[36] B. Li, W. Lu, S. Liu, and Z. Zhu, "Deep-learning-assisted network orchestration for on-demand and cost-effective vNF service chaining in inter-DC elastic optical networks," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 10, no. 10, pp. 29–41, 2018.

[37] F. Carpio, S. Dhahri, and A. Jukan, "VNF placement with replication for load balancing in NFV networks," in *Proc. IEEE Int. Commun. Conf.*, Paris, France, May 2017, pp. 1–6.

[38] B. T. Jou, O. Vidal, J. Cahill *et al.*, "Architecture options for satellite integration into 5G networks," in *Proc. Eur. Conf. Netw. Commun.*, Ljubljana, Slovenia, Jun. 2018, pp. 398–399.

[39] R. Ferrus, H. Koumaras, O. Sallent *et al.*, "On the virtualization and dynamic orchestration of satellite communication services," in *Proc. IEEE Veh. Technol. Conf.*, Montreal, Canada, Sep. 2016, pp. 1–5.

[40] T. Ahmed, R. Ferrus, R. Fedrizzi *et al.*, "Satellite gateway diversity in SDN/NFV-enabled satellite ground segment systems," in *Proc.*

IEEE Int Conf. Commun. Workshops, Paris, France, May 2017, pp. 882–887.
[41] T. Ahmed, A. Alleg, R. Ferrus *et al.*, "On-demand network slicing using SDN/NFV-enabled satellite ground segment systems," in *Proc. IEEE NetSoft Conf. Workshops*, Montreal, Canada, Jun. 2018, pp. 242–246.
[42] Y. Liu, Y. Chen, Y. Jiao *et al.*, "A shared satellite ground station using user-oriented virtualization technology," *IEEE Access*, vol. 8, pp. 63 923–63 934, 2020.
[43] R. K. Ahuja, J. B. Orlin, S. Pallottino *et al.*, "A multi-exchange heuristic for the single-source capacitated facility location problem," *Manage. Sci.*, vol. 50, no. 6, pp. 749–760, 2004.

Xiangqiang Gao received the B.Sc. degree in school of electronic engineering from Xidian University and the M.Sc. degree from Xi'an Microelectrinics Technology Institute, Xi'an, China, in 2012 and 2015, respectively. He is currently pursuing the Ph.D. degree with the School of Electronic and Information Engineering, Beihang University, Beijing, China. His research interests include rateless codes, software defined network and network function virtualization.

Rongke Liu received the B.Sc. degree in electronic engineering and Ph.D. degree in information and communication engineering from Beihang University, Beijing, China, in 1996 and 2002, respectively. From 2006 to 2007, he was a visiting professor at Florida Institute of Technology, Florida. In August, 2015, he visited the university of Tokyo as a senior visiting scholar. He is a Full Professor with the School of Electronic and Information Engineering in Beihang University, specializing in the fields of information and communication engineering. He has authored or co-authored more than 100 papers in journals and conferences, and edited four books. His current research interests include multimedia computing and space information network. He is a Member of the IEEE and ACM. Dr. Liu was one of the winners of education ministry's New Century Excellent Talents supporting plan in 2012.

Aryan Kaushik is currently a Research Fellow in Communications and Radar Transmission at the Institute of Communications and Connected Systems, University College London, United Kingdom. He received PhD in Communications Engineering at the Institute for Digital Communications, School of Engineering, The University of Edinburgh, United Kingdom, in 2020. He received MSc in Telecommunications from The Hong Kong University of Science and Technology, Hong Kong, in 2015. He has held visiting research appointments at the Wireless Communications and Signal Processing Lab, Imperial College London, UK, from 2019-20, the Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg, Luxembourg, in 2018, and the School of Electronic and Information Engineering, Beihang University, China, from 2017-19. His research interests are broadly in signal processing, radar, wireless communications, millimeter wave and multi-antenna communications.