Cost-Efficient Virtual Network Function Graph (vNFG) Provisioning in Multi-Domain Elastic Optical Networks

Yixiang Wang, Ping Lu, Wei Lu, Zuqing Zhu, Senior Member, IEEE

Abstract—Acting as a promising technology to bring revolutionary changes to how the networks are architected, network function virtualization (NFV) leverages IT virtualization technologies to instantiate various types of virtual network functions (vNFs) flexibly and dynamically on commodity hardware, which can be easily found in a variety of NFV infrastructure points-of-presence, e.g., datacenters (DCs). In this work, we investigate how to cost-effectively provision vNFGs with arbitrary topologies in a multi-domain elastic optical network (MD-EON) that consists of two domains, i.e., the private and public ones. We first formulate an integer linear programming (ILP) model with the objective to minimize the total resource cost of vNFG provisioning, and show that it can solve the problem exactly. With the ILP model, we analyze the complexity of the problem and prove that it is an \( NP \)-hard one. Then, we leverage the minimum \( k \)-cut problem to design two time-efficient heuristics. The results from extensive simulations verify the performance of our proposed algorithms, and indicate that they can balance the IT and spectrum resource usages intelligently according to the resource price setting in multi-domain environment.

Index Terms—Network function virtualization (NFV), virtual network function graph (vNFG), multi-domain, elastic optical networks (EONs).

I. INTRODUCTION

RECENTLY, network function virtualization (NFV) enjoys increasing popularity because the deployment of new network services can be greatly expedited with it [1, 2]. Specifically, NFV aims to migrate network functions from expensive special-purpose hardware to software-defined elements by leveraging IT resource virtualization, i.e., processing traffic with virtual network functions (vNFs) [3–8]. Meanwhile, bandwidth-intensive emerging network services, such as video streaming, social TV, etc., are developing rapidly on the Internet. Hence, how to deploy these services in the Internet cost-effectively has become a hot research topic. As these services usually use various network functions to process data traffic, e.g., video data needs compression and transcoding while voice data requires noise suppression and sampling, realizing them with vNFs in datacenters (DCs) can improve the adaptivity and efficiency of service provisioning [6, 7].

Note that, each network service may require a set of vNFs and the connectivity among them can formulate an arbitrary graph [3]. For example, to realize the NFV-based network defense system in [9] or to achieve the multipath routing based load balancing in [10], vNFs should be grouped into vNF graphs (vNFGs) instead of simple service chains. This makes deploying network services equivalent to provisioning vNFGs in an inter-DC network, which could be challenging when the constraints on IT and bandwidth resources both need to be addressed [6, 7]. Previously, people have considered the problem of vNF placement in [4, 5], while the studies in [11–14] have investigated how to deploy vNFGs in the forms of chain or tree. However, to the best of our knowledge, the problem of provisioning vNFGs with arbitrary topologies in inter-DC networks under IT and bandwidth resource constraints has not been fully explored yet. Meanwhile, the capacity and flexibility of physical infrastructure can affect the performance of vNF provisioning in inter-DC networks significantly [12, 14]. This is because the dynamics of bandwidth-intensive network services would make the traffic flowing through vNFs in a vNFG exhibit high peak throughput and high burstiness [15]. Consequently, agile bandwidth management in the optical layer of inter-DC networks would be necessary for provisioning vNFGs efficiently, which can be realized by leveraging the technical advances on flexible-grid elastic optical networks (EONs) [16].

In a practical scenario, the service provider (SP) of vNFGs might have to utilize resources from a multi-domain environment [17, 18]. It is known that multi-domain EONs (MD-EONs) can solve the inter-operability issues resulting from network elements owned by different vendors, provide enhanced network scalability and extended service reach, and handle the situation where the optical switches are geographically-distributed and/or operated by different carriers [18, 19]. More importantly, an SP can build a multi-DCs system connected with an EON as its private domain, and when the computing/storage capacity in the private domain becomes insufficient, it can rent DCs and EON infrastructure from external public networks as a quick and elastic solution. This actually mimics the classic hybrid cloud architecture [20]. Therefore, it would be relevant to study how to provision vNFGs cost-effectively in MD-EONs, which, to the best of our knowledge, has not been addressed in literature before.

Fig. 1 provides an illustrative example on vNFG provisioning in an MD-EON. As shown in Fig. 1(a), the inter-DC network is built over an MD-EON that consists of two domains, i.e., the private and public ones. To provision the network services that request for the vNFGs in Fig. 1(b), the SP needs to instantiate the vNFs in the DCs and then establishes necessary lightpaths to connect them accordingly. Then,
it can obtain the provisioning results as illustrated in Fig. 1(c).

Note that, the IT and spectrum resources in the private domain are usually cheaper, and thus for improving cost-effectiveness, the SP should only turn to the public domain when the private one does not have sufficient resources to provision the vNFGs. Although the problem looks similar to the multi-domain virtual network embedding (VNE) problem that has already been studied in [17], they are fundamentally different. Basically, in multi-domain VNE, the virtual networks’ topologies would not change during the embedding. While in our problem, the actual substrate topology used to support a vNFG can change with the vNF placement. For instance, in Fig. 1(c), after we placing vNFs 1 and 2 on DC $D_5$ and vNFs 3 and 4 on DC $D_5$, the vNFG with a mesh topology is carried by a substrate chain. More specifically, the connections for vNF $1 \rightarrow$ vNF $2$ and vNF $4 \rightarrow$ vNF $3$ are supported with intra-DC communications within DC $D_5$ and DC $D_5$, respectively, while the rest of the connections are merged and supported with the inter-domain link between DCs $D_5$ and $D_5$. Hence, our problem is more sophisticated than multi-domain VNE.

This paper studies how to provision vNFs for network services cost-effectively in MD-EONs. We first formulate an integer linear programming (ILP) model with the objective to minimize the total resource cost of vNFG provisioning, and show that it solves the problem exactly. With the ILP, we prove the problem’s $\mathcal{NP}$-hardness. Then, we leverage the minimum $k$-cut problem [21] to design two time-efficient heuristics. The results from extensive simulations verify the performance of our proposed algorithms. In summary, the major contributions of this work are as follows:

- We formulate an ILP model to solve the problem of vNFG provisioning in MD-EONs exactly for minimizing the total resource cost.
- We analyze the complexity of the problem and prove that it is an $\mathcal{NP}$-hard one.
- We propose two time-efficient heuristics by leveraging the minimum $k$-cut problem and conduct simulations for offline and online vNFG provisioning to verify that the proposed heuristics can provide near-optimal results.

The rest of this paper is organized as follows. We provide a survey on the related work in Section II. Section III introduces the network model, formulates the ILP model, and analyzes the problem’s complexity. The heuristics are proposed in Section IV, and the algorithms’ performance is evaluated in Section V. Finally, Section VI summarizes the paper.

II. RELATED WORK

Previously, people have studied the service provisioning schemes in MD-EONs from different perspectives [18, 19, 22–25]. Zhu et al. [19] considered how to realize service provisioning with energy-aware regenerator allocation in MD-EONs, and they also tried to address the security issues due to cross-domain attacks in MD-EONs and designed differential routing and spectrum assignment (RSA) schemes for intra- and inter-domain requests in [22]. For system implementations, the studies in [18, 23, 24] leveraged the idea of software-defined networking (SDN) to realize the network control and management frameworks for MD-EONs. The authors of [25] listed the challenges brought by integrating NFV and SDN in multi-layer and multi-domain network environments. However, these studies did not address how to deploy vNFs with arbitrary topologies in MD-EONs.

In [3], the white paper of NFV suggests that network services can be realized by deploying individual vNFs or vNFGs in substrate networks. However, most of previous studies on NFV-related service provisioning only considered individual vNFs or vNF chains [4, 5, 11, 12], which are relatively simple topologies for vNFGs. In [4, 5], the resource allocation schemes for placing individual vNFs were investigated. Mehraghdam et al. [11] considered the formulation of vNFG chains in an operator’s network, while the vNF chaining schemes in packet/optical DCs were discussed in [12]. The tree-type vNFGs were considered in [13, 14] for realizing multicast services in software-defined packet networks and inter-DC EONs, respectively. Since this work studies how to provision vNFGs with arbitrary topologies in MD-EONs, our network model of vNFGs is more generic than those used in the aforementioned studies.

Note that, as vNFs usually run as the applications in ensembles of virtual machines (VMs) [5], vNF provisioning and VM placement are generally related. Previously, Duong-Ba et al. [26] considered the VM management (i.e., placement
and migration) in DCs with the objective of minimizing energy consumption and cross-network traffic. In [27], the authors proposed a cost-aware two-phase meta-heuristic to minimize the cost of VM placement in geographically-distributed DCs. Nevertheless, we can see that in [26, 27], the VNs are usually independent and do not have specified relations as described by vNFs. Hence, despite of the similarity, VM placement and vNF provisioning have different constraints.

Lastly, as we have explained before, vNF provisioning is fundamentally different from the VNE problems that have been investigated in [28–35]. The authors of [31] studied an energy-efficient VNF model, while the energy-efficient VNE schemes in IP over optical networks have been considered in [35]. However, since the network model of vNF provisioning is different and we consider a multi-domain environment, the VNE algorithms proposed in these existing studies cannot be applied to solve our problem. For instance, in VNE, the virtual networks’ topologies are determined before the embedding, while in vNF provisioning, the actual topologies of vNFs are only finalized after they have been provisioned.

III. PROBLEM FORMULATION

In this section, we describe the network model of vNF provisioning in MD-EONs, formulate an ILP model to solve the problem exactly, and provide formal analysis on the problem’s complexity.

A. Network Model

We model the MD-EON as a directed graph \( G(D,E) \), where \( D \) is the DC set, and \( E \) is the set of fiber links that connect the DCs. There are \( F \) frequency slots (FS’) on each fiber link \( e \in E \) [36], and the unit cost of FS usage on link \( e \) is \( \beta_e \). Each DC \( d \in D \) can belong to the private or public domain in the MD-EON, and the SP uses the multi-dimensional IT resources on it (i.e., CPU cycles, memory and storage) to instantiate vNFs. We denote the capacity of IT resources on DC \( d \) as \( C_d = \langle C_d^C, C_d^M, C_d^S \rangle \), where \( C_d^C \), \( C_d^M \) and \( C_d^S \) represent the DC’s capacities of CPU cycle, memory and storage, respectively. We assume that \( M \) is the set of vNFs that the SP can instantiate in the MD-EON to accomplish all the network services requested by end users. \( v_d^m \) denotes the cost of the IT resource usage for instantiating a type \( m \) vNF in DC \( d \). The costs of IT and spectrum resources in the private domain are lower than those in the public domain.

A network service may use different vNFs to process the data from end users and these vNFs can formulate a vNFG as examples shown in Fig. 1(b). Each vNFG can be denoted as a directed graph \( G^v = \{U, A^v\} \), where \( U \) is the set of requested vNFs and \( A^v = \{a_{k,n} \mid (k, n \in [1, |U|]) \} \) is the traffic matrix for these vNFs. Here, \( a_{k,n} = \frac{\lambda_{k,n}}{\sum_{k,n} \lambda_{k,n}} \) represents the bandwidth requirement in FS’ for the traffic from the \( k \)-th vNF to the \( n \)-th vNF in the vNFG, and \( b_{k,n} \) is the actual bit-rate requirement and \( B_d \) represents the capacity of an FS. To generalize the vNFG model, we assume that multiple type \( m \) vNFs can appear in a vNFG and use \( N^m \) to denote the number of type \( m \) vNFs that are needed by the vNFG, and thus the total number of requested vNFs can be calculated as

\[
|U| = \sum_{m \in M} N^m. \quad (1)
\]

B. ILP Formulation

To provision a vNFG in the MD-EON, we need to accomplish two tasks. One is to instantiate vNFs in the DCs, and the other is to set up lightpaths by allocating FS’ on fiber links to satisfy the bandwidth requirements among the vNFs. In the process, the resource usages in the DCs and on the links should not exceed their capacities, and the lightpaths should also satisfy the spectrum contiguous and continuity constraints. For each DC pair, we precalculate \( K \) shortest paths with the Yen’s algorithm [37] and get all the possible RSA solutions on each path based on the current network status. The RSA solutions on all the precalculated paths are included in \( \tilde{S} \) to build the RSA solution set in the MD-EON [38], which are used as the MILP’s input. In the following, we formulate an ILP model to solve the problem of cost-effective vNFG provisioning in the MD-EON exactly. We denote the ILP model as vNFG-ILP.

Notations:

- \( G(D,E) \): the substrate topology of the MD-EON.
- \( F \): the number of FS’ on each fiber link \( e \).
- \( \beta_e \): the unit cost of FS usage on link \( e \).
- \( M \): the set of vNF types that the MD-EON can support.
- \( C_d \): the capacity of multi-dimensional IT resources on DC \( d \), i.e., \( C_d = \langle C_d^C, C_d^M, C_d^S \rangle \).
- \( c^C_m \): the CPU cycle consumption of a type \( m \) vNF.
- \( c^M_m \): the memory consumption of a type \( m \) vNF.
- \( c^S_m \): the storage consumption of a type \( m \) vNF.
- \( v_d^m \): the cost of the IT resource usage for instantiating a type \( m \) vNF in DC \( d \).
- \( G^v \): the vNFG request, i.e., \( G^v = \{U, A^v\} \).
- \( S \): the set of precalculated RSA solutions in \( G(D,E) \).
- \( s^* \): the number of FS’ used on each link in solution \( s \in \tilde{S} \).
- \( \tilde{S} \): the extended RSA solution set based on \( S \).
- \( \tilde{S}_d, d \): the extended RSA solution set for lightpaths from \( d_1 \) to \( d_2 \), \( d_1, d_2 \in D \).
- \( I_{k,m} \): the indicator that equals 1 if the \( k \)-th vNF in vNFG \( G^v \) belongs to type \( m \) and 0 otherwise.
- \( \pi_{e,x} \): the indicator that equals 1 if solution \( s \in \tilde{S} \) uses FS’ on link \( e \), and 0 otherwise.
- \( \pi^*_{f,x} \): the indicator that equals 1 if solution \( s \in \tilde{S} \) on link \( e \) includes the \( f \)-th FS, and 0 otherwise.
- \( I_{e,f} \): the indicator that equals 1 if the \( f \)-th FS has not been used on link \( e \), and 0 otherwise.

Variables:

- \( n_e \): the integer variable that indicates the number of used FS’ on link \( e \).
- \( h^m_d \): the integer variable that indicates the number of type \( m \) vNFs instantiated in DC \( d \).
- \( y_{k,n} \): the boolean variable that equals 1 if the \( k \)-th vNF in vNFG \( G^v \) is instantiated in DC \( d \) and 0 otherwise.
- \( f_{k,n} \): the boolean variable that equals 1 if vNFG \( G^v \), the lightpath from the \( k \)-th vNF to the \( n \)-th vNF uses RSA solution \( s \in \tilde{S} \), and 0 otherwise.
Objectives: The objective is to minimize the total cost of resources used for provisioning the vNFG in the MD-EON, as

\[
\text{Minimize} \quad C_r = C_{NF} + C_{FS},
\]

where \( C_{NF} \) denotes the total cost of IT resource usage for instantiating vNFs and \( C_{FS} \) is the total cost of FS usage. The total number of type \( m \) vNFs instantiated in DC \( d \) can be calculated as

\[
h^m_d = \sum_{k=1}^{\left| U \right|} y_{k,d} \cdot I_{k,m}, \quad \forall m \in M, \ d \in D.
\]

Then, the total cost of IT resource usage is

\[
C_{NF} = \sum_{d \in D} \sum_{m \in M} h^m_d \cdot v^m_d.
\]

On the other hand, to calculate the total cost of FS usage, we introduce the concept of extended RSA solution set, which is leveraged to cover the situation in which more than one vNFs in a vNFG are instantiated in the same DC. Fig. 2 explains how to obtain the extended RSA solution set \( \hat{S} \) with the precalculated RSA solution set \( S \). Specifically, with the original graph in Fig. 2(a), we add a dummy node (i.e., grey nodes in Fig. 2(b)) aside each node and connect them with a dummy bidirectional link (i.e., dotted lines in Fig. 2(b)) on which the number of FS’ is \( +\infty \) and the unit cost of FS usage is \( 0 \). Then, we get an extended graph as shown in Fig. 2(b), and with each dummy node \( d' \) and its original node \( d \), we calculate several feasible RSA solutions between them to support the traffic among the vNFs that are instantiated in the same DC \( d \). By adding these new RSA solutions in the original solution set \( S \), we obtain the extended RSA solution set \( \hat{S} \). Then, the FS usage on each link can be calculated as

\[
n_e = \sum_{k,n} \sum_{s} f_{k,n,s} \cdot z_{e,n} \cdot s^*, \quad \forall e \in E.
\]

Since each newly-added RSA solution \( s \) only includes a dummy link with \( z_{e,n} = 0 \), the situation in which more than one vNFs in a vNFG are instantiated in the same DC is correctly represented. Finally, we have

\[
C_{FS} = \sum_{e \in E} n_e \cdot \beta_e.
\]

Constraints:

1) IT Resource Constraints:

\[
\sum_{m \in M} h^m_d \cdot c^m_{i} \leq C^m_{d}, \quad \forall d \in D,
\]

\[
\sum_{m \in M} h^m_d \cdot c^m_{o} \leq C^m_{d}, \quad \forall d \in D,
\]

\[
\sum_{m \in M} h^m_d \cdot c^m_{s} \leq C^m_{s}, \quad \forall d \in D.
\]

Eqs. (7)-(9) ensure that when vNFs are instantiated in a certain DC, the consumption on IT resources, i.e., CPU cycles, memory and storage, should not exceed the corresponding IT resource capacities of that DC.

2) Spectrum Resource Constraints:

\[
\sum_{s \in S} f_{k,n,s} \cdot s^* \geq a_{k,n}, \quad \forall k,n \in [1,|U|].
\]

\[
\sum_{k,n \in S} f_{k,n,s} \cdot z_{f,n} \leq l_{e,f}, \quad \forall e \in E, f \in F.
\]

Eq. (10) ensures that the selected RSA solution can satisfy the spectrum requirement, and Eq. (11) guarantees that each FS \( f \) on a link \( e \) in \( E \) can only be used once, i.e., satisfying the spectrum non-overlapping constraint.

3) vNF Provisioning Constraints:

\[
\sum_{d \in D} y_{k,d} = 1, \quad \forall k \in [1,|U|].
\]

Eq. (12) ensures that each of the vNFs in a vNFG is instantiated in one and only one DC.

\[
\sum_{s \in S} f_{k,n,s} = 1, \quad \forall k,n \in [1,|U|].
\]

Eq. (13) ensures that for vNFG \( G' \), the traffic from the \( k \)-th vNF to the \( n \)-th vNF should use a feasible solution \( s \) in the extended RSA solution set \( \hat{S} \).

\[
y_{k,d_1} + y_{n,d_2} - 2 < \sum_{s \in S_{d_1,d_2}} f_{k,n,s},
\]

\[
\forall k,n \in [1,|U|], \forall d_1,d_2 \in D.
\]

Eq. (14) ensures that the results of vNF provisioning and spectrum allocation are matched. Specifically, for vNFG \( G'' \), if the \( k \)-th and \( n \)-th vNFs are placed on DCs \( d_1 \) and \( d_2 \), respectively, the traffic from the \( k \)-th vNF to the \( n \)-th vNF should use a feasible RSA solution \( s \) in the extended RSA solution set \( \hat{S}_{d_1,d_2}, \) and vice versa.
C. Complexity Analysis

Theorem 1. The optimization described by the aforementioned vNFG-ILP model for cost-efficient vNFG provisioning in an MD-EON is \( \mathcal{NP} \)-hard.

Proof: We prove the \( \mathcal{NP} \)-hardness of the optimization with restriction, i.e., restricting away certain aspects of the problem until a known \( \mathcal{NP} \)-hard problem appears [39]. For a certain vNFG to be served, we first relax the restrictions on spectrum resource. Basically, we set the FS on each fiber link in the MD-EON as infinite and assume that the unit cost of FS usage on each link is zero (i.e., \( \beta_e = 0, \forall e \in E \)). Hence, the spectrum allocation results become irrelevant and the optimization becomes to instantiate the vNFs of the vNFG in the DCs to minimize the total cost of IT resource usage. Then, if we treat each vNF as an item with a three-dimensional weight (i.e., CPU cycle, memory and storage consumptions) and the cost to instantiate it in a DC is its value, the optimization is transformed into the three-dimensional knapsack problem, which is known to be \( \mathcal{NP} \)-hard [39]. Therefore, since the special/restricted case of the optimization is the general case of a known \( \mathcal{NP} \)-hard problem, we prove that the optimization described by vNFG-ILP is \( \mathcal{NP} \)-hard.

IV. HEURISTIC ALGORITHMS

Since the problem of cost-efficient vNFG provisioning in MD-EONs is \( \mathcal{NP} \)-hard, we design two time-efficient heuristics in this section by leveraging the minimum \( k \)-cut problem discussed in [21]. Before discussing the heuristics, we introduce a concept for assisting the algorithm design.

Definition. For an arbitrary vNFG \( G' \), a vNF-DC mapping scheme is a feasible partition of its set of requested vNFs \( U \).

Fig. 3 shows an example on obtaining a vNF-DC mapping scheme for a vNFG \( G' \) = \( \{U, A\} \). First of all, with the traffic matrix \( A \) in Fig. 3(a), we can formulate the original topology of the vNFG as shown in Fig. 3(b), which shows the log connectivity among the five requested vNFs. Then, if we decide to deploy vNFs 1 and 2 on one DC and put each of the remaining vNFs to an individual DC, the original topology gets transformed to the four-node topology in Fig. 3(c) and the corresponding vNF-DC mapping scheme is also shown there. Apparently, the vNF-DC mapping scheme is just a feasible partition of the set of requested vNFs \( U \).

Theorem 2. For a vNFG with \( N \) vNFs, the number of all the possible vNF-DC mapping schemes is at least \( 2^{N} - N \).

Proof: First of all, it is known that the total number of partitions of an \( N \)-element set is the Bell number with the formula as [40]

\[
B(N + 1) = \sum_{k=0}^{N} \binom{N}{k} \cdot B(k),
\]

which is not an explicit expression. Then, if we define \( B(N, k) \) as the number of partitions when the \( N \) vNFs are divided into \( k \in [1, N] \) non-empty subsets, we can express \( B(N) \) as

\[
B(N) = \sum_{k=1}^{N} B(N, k).
\]

Then, if we consider to get the set partition with the method that first chooses \( k \) “seed” vNFs to form the initial subsets and then inserts the remaining \( N - k \) vNFs into the \( k \) subsets, the following inequality can be obtained

\[
B(N, k) \geq \binom{N}{k}, \quad \{N, k : N \geq 2, k \in [2, N]\}.
\]

Then, by combining Eqs. (17)-(18), we have

\[
B(N) = \sum_{k=1}^{N} B(N, k) \geq 1 + \binom{N}{2} + \ldots + \binom{N}{N} = \sum_{k=0}^{N} \binom{N}{k} - \binom{N}{1} = 2^{N} - N.
\]

Hence, we prove the theorem and this suggests that even without considering the spectrum allocation to connect the deployed vNFs, the possible vNF-DC mapping schemes increase exponentially with the size of the vNFG.

A. Iterative Two-Phase Algorithm with Minimum 2-Cut

Basically, according to the network model described in Section III-A, the cost for provisioning a vNFG in an MD-EON consists of two components, i.e., the IT resource cost for instantiating the requested vNFs and the spectrum resource cost for setting up lightpaths among the vNFs. Apparently, the first cost component can be reduced by trying to deploy the requested vNFs in the private domain where the cost of IT resource usage is lower. The second cost component can be reduced with two methods. Firstly, if we can deploy multiple vNFs on a same DC, the traffic among them becomes intra-DC and thus does not consume any spectrum resource on fiber links. Secondly, since the unit cost of FS usage on links in the private domain is lower than that of other links in the MD-EON, the spectrum resource cost can also be reduced by adjusting the vNF-DC mapping scheme to use as few DCs in the public domain as possible. However, according to Theorem 2, the number of possible vNF-DC mapping schemes increases exponentially with the number of vNFs in a vNFG, and thus a polynomial-time algorithm should not try to traverse all the vNF-DC mapping schemes.

Based on these considerations, we propose an iterative two-phase vNFG provisioning algorithm that applies the minimum 2-cut (M2C) algorithm [21] to the vNFG repeatedly, i.e., iterative two-phase with M2C (iTP-M2C). The two phases are DC selection and vNFG partitioning, which are repeated until the vNFG is successfully provisioned. Specifically, the DC selection chooses an appropriate DC for vNF deployment and then the vNFG partitioning applies M2C to determine the subset of the vNFs to be put in that DC. Apparently, the selected DC(s) should have sufficient and cheap IT resources as well as be connected with fiber link(s) that have enough and
low-cost spectrum resources. Hence, we define the following cost metric for each DC
\[ w_d = \alpha \frac{1}{|M|} \sum_{m \in M} \frac{v^m_d}{h^m_d} + \gamma \cdot \sum_{\{e=(d,v) \in E\}} \beta_e S_e, \tag{20} \]
where \( \alpha \) and \( \gamma \) are the coefficients to balance the importance of IT and spectrum resources in the DC selection, \( v^m_d \) is the price of the IT resource usage for instantiating a type \( m \) vNF in DC \( d \), \( h^m_d \) is the number of type \( m \) vNFs that can be instantiated in DC \( d \), and for each link \( e \) that originates from or ends at DC \( d \), \( \beta_e \) is the unit cost of FS usage on it and \( S_e \) denotes the number of available FS’ on it.

Basically, after determining the cost metrics for all the DCs in the MD-EON, we choose the DC with the smallest cost metric in the DC selection phase. Then, in the phase of vNFG partitioning, we first apply the M2C algorithm to divide the vNFG into two sub-graphs that require the minimum FS’ for communication in between, and then adjust the two sub-graphs with least spectrum requirement increase each time until the vNFs in one of the sub-graphs can be accommodated in the selected DC. The operations in these two phases are repeated until the vNFG is provisioned or blocked.

Algorithm 1 shows the detailed procedure of iTP-M2C. Line 1 is for the initialization. The while-loop that covers Lines 2-42 applies the aforementioned two phases iteratively until the vNFG is provisioned or blocked. Here, \( D_{temp} \) stores the DCs that currently can still be used to instantiate vNFs, and Lines 3-5 will block the vNFG request \( G' \) if \( D_{temp} \) is empty. Otherwise, Lines 7-8 perform the DC selection based on the cost metric defined in Eq. (20). Then, if all the outstanding vNFs in \( U_{temp} \) can be accommodated in the selected DC \( d \), Lines 10-20 check whether the spectrum resources in the MD-EON are enough to set up all the necessary inter-DC lightpaths. If yes, Lines 21-23 mark \( G' \) as provisioned with the selected DCs in \( D_{used} \). Otherwise, we block \( G' \) as shown in Lines 15-17. On the other hand, if all the outstanding vNFs in \( U_{temp} \) cannot be accommodated in the selected DC \( d \), Lines 25-26 apply the M2C algorithm to divide the outstanding vNFs into two subsets (i.e., \( U_1 \) and \( U_2 \)) that have the least traffic in between, and then adjust the two subsets with the least traffic increase in between until the vNFs in one of the subsets (w.o.l.g., we assume that it is \( U_1 \)) can be accommodated in the selected DC \( d \). Again, Lines 27-37 try to set up all the necessary inter-DC lightpaths with the lowest spectrum resource cost. If this can be done, Lines 38-39 update the variables and proceed to the next iteration.

Four subroutines contribute to the time complexity of iTP-M2C, i.e., cost metric calculation and DC sorting, M2C operation, subset adjustment, RSA solution selection and spectrum refreshing. The complexity of the first one is \( O(|D| \cdot \log(|D|)) \). We leverage the K-means clustering discussed in [41] to realize M2C, and its complexity is \( O(|U|) \). The complexity of the subset adjustment is also \( O(|U|) \). There are \( K \) precalculated paths between any two DCs and \( F \) FS’ on each link, so the complexity of RSA solution selection and spectrum refreshing is \( O((|D| - 1) \cdot (K \cdot F + E)) \) in the worst case. Finally, we would use at most \(|D| \) iterations to serve a vNFG request, and thus the overall time complexity of iTP-M2C is \( O(|D|^2 \cdot \log(|D|) + K \cdot F + E) + |D| \cdot |U|) \).

B. Two-Phase Algorithm with Minimum k-Cut

Note that, since iTP-M2C uses a greedy strategy to minimize the total cost of vNFG provisioning in each iteration, it might not be able to obtain the solution that is global optimal. Therefore, we propose another two-phase algorithm that utilizes minimum k-cut (MkC) to search for low-cost vNFG provisioning schemes, which is referred to as two-phase algorithm with MkC (TP-MkC). Specifically, TP-MkC first tries to divide the vNFs in \( G' = \{U, A\} \) into \( k = 1, 2, \ldots, \min(|D|, |U|) \) subsets with the MkC algorithm that ensures the minimum traffic among them. Then, the original vNFG is transformed into \( \min(|D|, |U|) \) different virtual networks (VN), and we try to embed the VN into the MD-EON with consideration of the DC cost metric defined in Eq. (20) and spectrum resource cost. After obtaining all the feasible embedding schemes, we calculate the total costs of them and then select the one with the lowest cost to provision the vNFG. Algorithm 2 provides the detailed procedure of TP-MkC. Lines 1-2 are for the initialization. The for-loop covering Lines 3-11 divides the vNFs in \( G' = \{U, A\} \) into \( k = 1, 2, \ldots, \min(|D|, |U|) \) subsets with the MkC constructs the corresponding VN, and then tries to embed each VN into
Algorithm 1: Iterative Two-Phase Algorithm with M2C

**input**: vNFG request \( G' = \{U, A\} \), and MD-EON \( G(D, E) \).

1. \( D_{\text{temp}} = D, \ D_{\text{used}} = \emptyset, \ U_{\text{temp}} = U; \)
2. while \( U_{\text{temp}} \neq \emptyset \) do
   3. if \( D_{\text{temp}} = \emptyset \) then
      4. mark \( G' \) as blocked;
      5. return;
   6. else calculate cost metric \( w_d \) with Eq. (20) for each DC \( d \in D_{\text{temp}} \) based on status of \( G(D, E) \);
   7. choose the DC \( d \) with the minimum \( w_d \);
   8. if vNFs in \( U_{\text{temp}} \) can be deployed in DC \( d \) then
      9. if \( D_{\text{used}} \neq \emptyset \) then
         10. for each DC \( d' \in D_{\text{used}} \) do
            11. try to find an RSA solution with the lowest spectrum cost to support the traffic among the vNFs in DC \( d' \) and those in DC \( d \);
            12. if the RSA solution can be found then
               13. update network resource usage;
            else
               14. mark \( G' \) as blocked;
               15. return;
            end
         end
      16. \( D_{\text{used}} = D_{\text{used}} \cup d; \)
      17. mark \( G' \) as provisioned with DCs in \( D_{\text{used}} \);
      18. end
   19. else apply M2C to divide the vNFs in \( U_{\text{temp}} \) into two subsets \( U_1 \) and \( U_2 \) with the least traffic between them;
   20. adjust \( U_1 \) and \( U_2 \) with the least traffic increase in between such that the vNFs in \( U_1 \) can be accommodated in DC \( d \);
   21. if \( D_{\text{used}} \neq \emptyset \) then
      22. for each DC \( d' \in D_{\text{used}} \) do
         23. try to find an RSA solution with the lowest spectrum cost to support the traffic among the vNFs in DC \( d' \) and those in DC \( d \);
         24. if the RSA solution can be found then
            25. update network resource usage;
         else
            26. mark \( G' \) as blocked;
            27. return;
         end
      end
   28. \( D_{\text{used}} = D_{\text{used}} \cup d, \ D_{\text{temp}} = D_{\text{temp}} \setminus d; \)
   29. \( U_{\text{temp}} = U_{\text{temp}} \setminus U_1; \)
   30. end
end

\( G(D, E) \) with node mapping based on DC cost metric \( \{ w_d \} \) and spectrum allocation based on FS cost for minimizing the total provisioning cost. If a feasible embedding scheme can be obtained for a VN, we store it in the solution set \( \Omega \) and calculate the corresponding total provisioning cost, as shown in Lines 7-10. Finally, after trying all the VNs, we select the embedding scheme with the lowest total cost to provision \( G' \) as illustrated in Lines 12-18. The time complexity of TP-M\&C is \( O(|D| \cdot \log(|D|) + \min(|D|, |U|) \cdot (|U| + |D|^2 \cdot (K \cdot F + E))) \).

Algorithm 2: Two-Phase Algorithm with M\&C

**input**: vNFG request \( G' = \{U, A\} \), and MD-EON \( G(D, E) \).

1. \( U_{\text{temp}} = \emptyset, \ \Omega = \emptyset; \)
2. calculate cost metric \( w_d \) with Eq. (20) for each DC \( d \in D \) based on status of \( G(D, E) \);
3. for \( k = 1 \) to \( \min(|D|, |U|) \) do
   4. apply M\&C to divide the vNFs in \( U \) into \( k \) subsets with the least traffic among them;
   5. store the \( k \) subsets in \( U_{\text{temp}} \) and construct a VN based on them;
   6. try to map the VN into \( G(D, E) \) with node mapping based on \( \{ w_d \} \) and spectrum allocation based on FS cost;
   7. if the VN can be embedded successfully then
      8. store the feasible embedding scheme in \( \Omega \);
      9. calculate total resource cost of the scheme;
   end
11 end
12 if \( \Omega \neq \emptyset \) then
13 select the embedding scheme with the lowest total cost to provision \( G' \);
14 mark \( G' \) as provisioned;
15 update network resource usage;
16 else
17 mark \( G' \) as blocked;
18 end

V. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the proposed algorithms, i.e., the vNFG-ILP, iTP-M2C, and TP-M\&C with the six-node topology in Fig. 1(a) and the NSFNET topology in Fig. 4. Note that, in addition to these two, there are also other practical multi-domain topologies that can be leveraged to model the MD-EON [42]. Our simulations consider both offline and online scenarios. Here, the offline scenario means that all the vNFGs are known in advance. While for the online scenario, the vNFGs can come and leave on-the-fly following the Poisson traffic model, i.e., the average arrival rate is \( \lambda \) vNFGs per time-unit and the average holding time of each vNFG is \( \frac{1}{\mu} \) time-units. Hence, the traffic load is \( \frac{\lambda}{\mu} \) in Erlangs.

A. Performance Metrics

We first explain the performance metrics that are used in the simulations. The cost of the IT and spectrum resources
used for provisioning vNFGs is an important metric to evaluate whether an algorithm can achieve high cost-efficiency. Therefore, we obtain the average cost as the total resource cost averaged by the number of vNFGs. We also consider the blocking probability of vNFGs in the online scenario. Note that, to evaluate the time efficiency of the algorithms, we present the results on the total running time for the offline scenario as well. Moreover, we design a benchmark algorithm with the straightforward approach, i.e., it selects the DC with the most available IT resources to instantiate vNFs and adopts the same method as iTP-M2C to support the traffic among vNFs provisioned in different DCs iteratively. The benchmark algorithm is named as the first-fit algorithm (FF). To ensure sufficient statistic accuracy, we obtain each data point by averaging the results from 100 independent simulations.

B. Evaluations for Offline Scenario

We consider three types of vNFs. A vNF 1 needs 1 unit of CPU cycles, 3.75 units of memory and 4 units of storage, a vNF 2 needs 2 units of CPU cycles, 7.5 units of memory and 32 units of storage, and a vNF 3 needs 4 units of CPU cycles, 15 units of memory and 80 units of storage [43]. The costs of the IT resource usage for instantiating a vNF 1, a vNF 2 and a vNF 3 are 0.067, 0.133, and 0.266 units, respectively, in the private domain, and the costs are doubled in the public domain [43]. There are three types of fiber links in the MD-EON, i.e., the intra-links in the private domain, the inter-links between the private and public domains, and the intra-links in the public domain, the unit costs of FS usage on them are set as 0.005, 0.01, and 0.02 units, respectively [43].

Firstly, we fix the scale of vNFG requests and change the number of requests in each group to observe the performance of the algorithms. Here for each vNF, we assume that it randomly requires [2, 3] instances of vNF 1, [0, 2] instances of vNF 2, and [0, 1] instances of vNF 3. Table I shows the simulation results with the six-node topology. As expected, vNFG-ILP provides the lowest average cost all the time, which verifies that it can get the optimal provisioning schemes for the vNFGs. In terms of the average cost, vNFG-ILP is followed by TP-M\(k\)C. The fact that the average costs from TP-M\(k\)C are the same as or only slightly higher than those from vNFG-ILP verifies its effectiveness on providing near-optimal solutions. The performance of iTP-M2C is worse than that of TP-M\(k\)C since it uses a greedy strategy to minimize the total cost of vNFG provisioning in each iteration. Among all the algorithms, FF performs the worst in terms of the average cost. Meanwhile, although vNFG-ILP can obtain the optimal solutions, its running time is also the longest and thus it can easily become intractable for large-scale problems. The running time of TP-M\(k\)C is much shorter than vNFG-ILP, followed by iTP-M2C and FF.

We then perform simulations with different vNFG sizes (i.e., number of vNFs in each vNFG) and fix the number of vNFG requests in each group as 10. Table II shows the results on the average cost and running time. As we can see that vNFG-ILP always outperforms the other three algorithms in terms of the average cost, but its running time increases rapidly with the size of the vNFGs and it cannot finish the problem-solving within a reasonable time period, when the size of the vNFGs is larger than 12. TP-M\(k\)C still provides lower average cost than iTP-M2C and FF, and FF still performs the worst in terms of the average cost. The running time of these four algorithms also follows a similar trend as that in Table I. Note that, although the average running time per vNFG of TP-M\(k\)C is longer than that of iTP-M2C and FF, it can finish the computation within 0.5 second for provisioning a vNFG whose size is 12, which is still reasonably good for practical network operation.

C. Evaluations for Online Scenario

For the online scenario, the vNFG requests can be blocked due to the insufficiency of spectrum resources on fiber links, multi-dimensional IT resources in DCs, or both. This time, we only consider the NSFNET topology. On each DC, the CPU, memory and storage resources range randomly within [50, 120] units, [150, 400] units, and [500, 1500] units, respectively. Each fiber link carries 1000 FS’. For each vNF, we assume that it randomly requires [2, 4] instances of vNF 1, [0, 4] instances of vNF 2, and [0, 2] instances of vNF 3. The rest of the simulation parameters are the same as those in the offline scenario. Each simulation runs for 10,000 time units.

1) Basic Performance Comparison: Fig. 5 shows the simulation results on blocking probability. We observe that the blocking performance of TP-M\(k\)C is the best, followed by iTP-M2C, while FF performs the worst. This is because FF does not consider the traffic demands among the vNFs when instantiating them, which would cause unnecessary and unbalanced spectrum utilization in the MD-EON and eventually lead to the blocking of vNFGs. On the other hand, since iTP-M2C and TP-M\(k\)C properly address the traffic demands when designing the vNF-DC mapping schemes, they achieve better utilization of the spectrum resources.

Fig. 6 plots the results on average total resource cost. It is interesting to notice that FF also provides the highest average resource cost, even though its blocking performance is the worst. This further confirms the cost-efficiency of iTP-M2C and TP-M\(k\)C. When comparing the results from iTP-M2C and TP-M\(k\)C, we can see that the average resource cost from TP-M\(k\)C is lower than that from iTP-M2C only when the traffic load is lower than 60 Erlangs. This is because since the blocking probability of iTP-M2C is higher than that of TP-M\(k\)C, TP-M\(k\)C would serve more requests using resources at
TABLE I
RESULTS WITH FIXED VNFG SCALE UNDER SIX-NODE TOPOLOGY FOR OFFLINE SCENARIO.

<table>
<thead>
<tr>
<th>Number of VNFGs</th>
<th>Average cost per VNFG (units)</th>
<th>Average running time per VNFG (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>vNFG-ILP</td>
<td>iTP-M2C</td>
</tr>
<tr>
<td>10</td>
<td>0.355</td>
<td>0.544</td>
</tr>
<tr>
<td>20</td>
<td>0.625</td>
<td>0.652</td>
</tr>
<tr>
<td>30</td>
<td>0.686</td>
<td>0.654</td>
</tr>
<tr>
<td>40</td>
<td>0.594</td>
<td>0.645</td>
</tr>
<tr>
<td>50</td>
<td>0.621</td>
<td>0.649</td>
</tr>
<tr>
<td>60</td>
<td>0.559</td>
<td>0.616</td>
</tr>
</tbody>
</table>

TABLE II
RESULTS WITH FIXED NUMBER OF VNFGS UNDER SIX-NODE TOPOLOGY FOR OFFLINE SCENARIO.

<table>
<thead>
<tr>
<th>Size of VNFGs</th>
<th>Average cost per VNFG (units)</th>
<th>Average running time per VNFG (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>vNFG-ILP</td>
<td>iTP-M2C</td>
</tr>
<tr>
<td>4</td>
<td>0.324</td>
<td>0.606</td>
</tr>
<tr>
<td>8</td>
<td>1.270</td>
<td>1.436</td>
</tr>
<tr>
<td>12</td>
<td>1.963</td>
<td>2.186</td>
</tr>
</tbody>
</table>

Fig. 5. Blocking probability for online scenario.

Fig. 6. Average total resource cost for online scenario.

Fig. 7. Average IT cost for online scenario.

Fig. 8. Average spectrum cost for online scenario.

...
private and public domains (Inter), and the intra-link in the public domain (Public). It can be seen that when the traffic load is low, iTP-M2C and TP-M/C try to use the network resources in the private domain. Only when the traffic load increases and the private domain can not serve the requests, they consider to use the network resources out of the private domain, because iTP-M2C and TP-M/C arrange the IT and spectrum resource utilizations in a more intelligent way to better balance the costs of IT and spectrum consumptions. However, FF uses almost equal resources in the private and public domains for the reason that it only considers the quantity of resource in each DC. These results verify that our algorithms can properly address different resource prices in the MD-EON to achieve cost-efficient vNFG provisioning.

3) Impact of Resource Price Setting: Finally, we run simulations with different resource price settings to further confirm the robustness of our proposed algorithms. Firstly, we keep the price setting on spectrum resources unchanged but increase the prices of the IT resources in the public domain to four times of those in the private domain. This simulation scenario is referred to as the “high IT price setting”. Fig. 9 shows the results on blocking probability, which exhibit the similar trend as that in Fig. 5. Table IV presents the results on the average total resource cost, average IT cost and average spectrum cost. We observe that because the price of IT resources in the public domain is higher, the average IT cost increases a lot for all the algorithms. However, the general trends among the algorithms for the costs are the same as those in Figs. 6-8. Table V shows the distributions of resource utilization in the MD-EON in Table VII also show the similar trend as that in Fig. 10. Finally, the results on the distributions of resource utilization in the MD-EON in Table VII also show that iTP-M2C and TP-M/C can adjust the resource utilizations intelligently based on the resource prices.

Note that, in addition to the aforementioned scenarios, we also simulate other scenarios to further verify the performance of iTP-M2C and TP-M/C, i.e., changing the vNFs with random topologies to pure vNF chains or vNF trees and changing the MD-EON’s topology to the practical ones in [42]. Since the simulation results still exhibit the similar trends as discussed above, we omit them due to the page limit.

VI. CONCLUSIONS

In this paper, we studied how to provision vNFGs cost-effectively in an MD-EON that consists of two domains,
<table>
<thead>
<tr>
<th>ERLangs</th>
<th>FF</th>
<th>tTP-M2C</th>
<th>tTP-MC</th>
<th>FF</th>
<th>tTP-M2C</th>
<th>tTP-MC</th>
<th>FF</th>
<th>tTP-M2C</th>
<th>tTP-MC</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU cycles</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>20</td>
<td>1.17x+5</td>
<td>1.11x+5</td>
<td>1.15x+5</td>
<td>1.09x+5</td>
<td>1.09x+5</td>
<td>1.08x+5</td>
<td>8.93x+4</td>
<td>8.59x+4</td>
</tr>
<tr>
<td>Private</td>
<td>42.8%</td>
<td>83.6%</td>
<td>99.3%</td>
<td>47.7%</td>
<td>58.4%</td>
<td>63.6%</td>
<td>47.7%</td>
<td>52.8%</td>
<td>52.5%</td>
</tr>
<tr>
<td>Public</td>
<td>57.2%</td>
<td>16.8%</td>
<td>0.9%</td>
<td>52.2%</td>
<td>41.6%</td>
<td>36.4%</td>
<td>52.2%</td>
<td>47.2%</td>
<td>47.5%</td>
</tr>
<tr>
<td>Memory</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>4.17x+5</td>
<td>4.17x+5</td>
<td>4.17x+5</td>
<td>4.05x+5</td>
<td>4.09x+5</td>
<td>4.10x+5</td>
<td>3.15x+5</td>
<td>3.22x+5</td>
<td>3.30x+5</td>
</tr>
<tr>
<td>Private</td>
<td>42.8%</td>
<td>83.6%</td>
<td>99.3%</td>
<td>47.7%</td>
<td>58.4%</td>
<td>63.6%</td>
<td>47.7%</td>
<td>52.8%</td>
<td>52.5%</td>
</tr>
<tr>
<td>Public</td>
<td>57.2%</td>
<td>16.8%</td>
<td>0.9%</td>
<td>52.2%</td>
<td>41.6%</td>
<td>36.4%</td>
<td>52.2%</td>
<td>47.2%</td>
<td>47.5%</td>
</tr>
<tr>
<td>Storage</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>1.58x+6</td>
<td>1.58x+6</td>
<td>1.58x+6</td>
<td>1.58x+6</td>
<td>1.58x+6</td>
<td>1.58x+6</td>
<td>1.58x+6</td>
<td>1.58x+6</td>
<td>1.58x+6</td>
</tr>
<tr>
<td>Private</td>
<td>42.8%</td>
<td>83.6%</td>
<td>99.3%</td>
<td>47.7%</td>
<td>58.4%</td>
<td>63.6%</td>
<td>47.7%</td>
<td>52.8%</td>
<td>52.5%</td>
</tr>
<tr>
<td>Public</td>
<td>57.2%</td>
<td>16.8%</td>
<td>0.9%</td>
<td>52.2%</td>
<td>41.6%</td>
<td>36.4%</td>
<td>52.2%</td>
<td>47.2%</td>
<td>47.5%</td>
</tr>
<tr>
<td>Spectrum</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>157</td>
<td>0</td>
<td>0</td>
<td>3.20x+4</td>
<td>2.17x+4</td>
<td>1.43x+4</td>
<td>4.64x+4</td>
<td>5.65x+4</td>
<td>6.18x+4</td>
</tr>
<tr>
<td>Private</td>
<td>17.8%</td>
<td>—</td>
<td>—</td>
<td>38.2%</td>
<td>55.3%</td>
<td>55.0%</td>
<td>38.7%</td>
<td>44.4%</td>
<td>43.6%</td>
</tr>
<tr>
<td>Inter</td>
<td>17.8%</td>
<td>—</td>
<td>—</td>
<td>28.6%</td>
<td>30.8%</td>
<td>30.4%</td>
<td>26.8%</td>
<td>28.3%</td>
<td>28.1%</td>
</tr>
<tr>
<td>Public</td>
<td>64.4%</td>
<td>—</td>
<td>—</td>
<td>33.2%</td>
<td>15.9%</td>
<td>14%</td>
<td>34.5%</td>
<td>28.8%</td>
<td>28.5%</td>
</tr>
</tbody>
</table>

**TABLE IV**

<table>
<thead>
<tr>
<th>ERLangs</th>
<th>FF</th>
<th>tTP-M2C</th>
<th>tTP-MC</th>
<th>FF</th>
<th>tTP-M2C</th>
<th>tTP-MC</th>
<th>FF</th>
<th>tTP-M2C</th>
<th>tTP-MC</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU cycles</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>20</td>
<td>1.11x+5</td>
<td>1.11x+5</td>
<td>1.17x+5</td>
<td>1.08x+5</td>
<td>1.09x+5</td>
<td>1.09x+5</td>
<td>8.39x+4</td>
<td>8.56x+4</td>
</tr>
<tr>
<td>Private</td>
<td>42.3%</td>
<td>83.6%</td>
<td>99.3%</td>
<td>41.1%</td>
<td>58.4%</td>
<td>63.6%</td>
<td>35.1%</td>
<td>52.8%</td>
<td>52.5%</td>
</tr>
<tr>
<td>Public</td>
<td>57.7%</td>
<td>16.8%</td>
<td>0.9%</td>
<td>52.2%</td>
<td>41.6%</td>
<td>36.4%</td>
<td>52.2%</td>
<td>47.2%</td>
<td>47.5%</td>
</tr>
<tr>
<td>Memory</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>4.17x+5</td>
<td>4.17x+5</td>
<td>4.17x+5</td>
<td>4.05x+5</td>
<td>4.09x+5</td>
<td>4.10x+5</td>
<td>3.15x+5</td>
<td>3.22x+5</td>
<td>3.30x+5</td>
</tr>
<tr>
<td>Private</td>
<td>42.3%</td>
<td>83.6%</td>
<td>99.3%</td>
<td>41.1%</td>
<td>58.4%</td>
<td>63.6%</td>
<td>35.1%</td>
<td>52.8%</td>
<td>52.5%</td>
</tr>
<tr>
<td>Public</td>
<td>57.7%</td>
<td>16.8%</td>
<td>0.9%</td>
<td>52.2%</td>
<td>41.6%</td>
<td>36.4%</td>
<td>52.2%</td>
<td>47.2%</td>
<td>47.5%</td>
</tr>
<tr>
<td>Storage</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>1.58x+6</td>
<td>1.58x+6</td>
<td>1.58x+6</td>
<td>1.58x+6</td>
<td>1.58x+6</td>
<td>1.58x+6</td>
<td>1.58x+6</td>
<td>1.58x+6</td>
<td>1.58x+6</td>
</tr>
<tr>
<td>Private</td>
<td>42.3%</td>
<td>83.6%</td>
<td>99.3%</td>
<td>41.1%</td>
<td>58.4%</td>
<td>63.6%</td>
<td>35.1%</td>
<td>52.8%</td>
<td>52.5%</td>
</tr>
<tr>
<td>Public</td>
<td>57.7%</td>
<td>16.8%</td>
<td>0.9%</td>
<td>52.2%</td>
<td>41.6%</td>
<td>36.4%</td>
<td>52.2%</td>
<td>47.2%</td>
<td>47.5%</td>
</tr>
<tr>
<td>Spectrum</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>157</td>
<td>7</td>
<td>0</td>
<td>3.20x+4</td>
<td>2.17x+4</td>
<td>1.43x+4</td>
<td>4.64x+4</td>
<td>5.65x+4</td>
<td>6.18x+4</td>
</tr>
<tr>
<td>Private</td>
<td>17.8%</td>
<td>0</td>
<td>0</td>
<td>38.2%</td>
<td>55.3%</td>
<td>55.0%</td>
<td>38.7%</td>
<td>44.4%</td>
<td>43.6%</td>
</tr>
<tr>
<td>Inter</td>
<td>17.8%</td>
<td>0</td>
<td>0</td>
<td>28.6%</td>
<td>30.8%</td>
<td>30.4%</td>
<td>26.8%</td>
<td>28.3%</td>
<td>28.1%</td>
</tr>
<tr>
<td>Public</td>
<td>64.4%</td>
<td>0</td>
<td>0</td>
<td>33.2%</td>
<td>15.9%</td>
<td>14%</td>
<td>34.5%</td>
<td>28.8%</td>
<td>28.5%</td>
</tr>
</tbody>
</table>

i.e., the private and public ones. Because the network model and optimization objective of our problem are different from those of existing studies on VNE and vNF chain provisioning, the algorithms designed in them cannot be applied. We first formulated an ILP model with the objective to minimize the total resource cost of vNFG provisioning, and showed that it can solve the problem exactly. With the ILP model, we analyzed the complexity of the problem and proved its \(N^P\)-hardness. Then, we leveraged the minimum \(k\)-cut problem to design two time-efficient heuristics. The results from extensive simulations verified that our proposed algorithms outperformed an existing benchmark in terms of blocking probability and average total cost in different simulation scenarios. More specifically, our algorithms arranged the IT and spectrum resources intelligently to balance their usages according to the price setting in the MD-EON.

**ACKNOWLEDGMENTS**

This work was supported in part by the NSFC Project 61371117, Natural Science Research Project for Universities in Anhui (KJ2014ZD38), and the SPR Program of CAS (XDA06011202).

**REFERENCES**


