IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 22, 2020, accepted March 9, 2020, date of publication March 19, 2020, date of current version March 31, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2982073

Energy Optimized Virtual Network Embedding With Location Constraint in the Enterprise Network

XIN CONG[1], **KAI SHUANG**[2], **AND LINGLING ZI**[1]
[1]School of Electronic and Information Engineering, Liaoning Technical University, Huludao 125105, China
[2]State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

Corresponding author: Kai Shuang (shuangk@bupt.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61602227 and Grant 61702241, in part by the China Scholarship Council, and in part by the National Key Research and Development Program of China under Grant 2016QY01W0200.

ABSTRACT To satisfy enterprise demands of analyzing and dealing with the large scale of data with lower costs, an effective method is to integrate the servers and computers and use virtualization technology to construct an enterprise network. Prior studies on network virtualization have mainly been executed in the cloud; however, these studies may not be appropriate for enterprise networks for two reasons: i) the goal of most of them is to generate more revenues for cloud providers, but focus less on saving costs; ii) the physical machines are relatively concentrated in the cloud platform but dispersed over different geographic locations in enterprise networks. In this paper, we solve the problem of energy-optimized virtual network embedding with location constraints (EO-VNE). First, the node and link capabilities in enterprise networks are defined in the form of complex number theory, unifying computers and virtual requests. Second, the normalized method of computing and storage capabilities are proposed to identify the node capability. Third, an energy model of the enterprise network is built, and using this model, EO-VNE is shown to be NP-complete. Finally, an energy-optimized virtual network embedding with a location constraint algorithm (EOLC) is proposed to minimize the energy consumption under the constraint of node position. The experiments show that EOLC consumes less energy compared with the algorithm of energy-aware virtual network embedding with dynamic demands (EAD). It also has better performance than the location constraint algorithm based on bisection (GLC).

INDEX TERMS Energy optimization, location-constrained, virtual network embedding, enterprise network.

I. INTRODUCTION

Generally, enterprises require a large infrastructure to analyze and process large-scale data. To obtain the infrastructure, they can rent cloud platforms or build private data centers. However, they are willing to construct their private centers to protect the privacy of data, which may face high building costs. To decrease this cost, a good approach is to utilize the existing computing devices and enterprise network to compose the substrate networks that can accommodate and run data tasks. In this case, an inevitable problem is to efficiently share the resources, named virtual network embedding (VNE) [1]. VNE allows heterogeneous networks to coexist in the same substrate network, and the resources in VNE can be shared by different types of tasks [2].

The VNE problem has been extensively studied in cloud platforms [3]–[5]. These studies assume good conditions, such as infinite network resources and sufficient task execution time. Additionally, most of these studies are aimed at optimizing the benefits of cloud providers. However, the above conditions do not apply to the enterprise network for the following reasons. 1) Because computers in the enterprise network are located in different offices and they can only provide limited resources, a virtual machine can only be mapped on one physical machine in a single process of VNE. Additionally, a communication link between two virtual nodes can only be embedded on a single physical path. 2) The arriving number of virtual requests (VRs) varies at different time, so dynamic features must be considered during the procedure of VNE. 3) Because computers in the enterprise network provide various resources and QoS, VRs may choose different computers to obtain suitable

The associate editor coordinating the review of this manuscript and approving it for publication was Jenny Mahoney.

services. This implies that the location constraint of physical nodes should be considered. 4) The enterprise network generates the tasks to be executed, so benefit optimization is not the goal of VNE, but cost savings are its main goal. Simultaneously, energy consumption is a major component of cost savings. The problem in this paper is VNE energy optimization with location-constrained for the enterprise network (EO-VNE).

To solve the above problem, an energy optimization virtual network embedding with a location constraint algorithm (EOLC) is proposed for the enterprise network. The main contributions of our paper are described as follows.

(1) The complex number theory is adopted to define the node capability and link capability in the network, which, to the best of our knowledge, has not been studied before. Based on the complex number theory, the computing and storage capabilities are integrated into the real part, reducing the times of comparison whether a physical node can accommodate the virtual requests.

(2) A dynamic feature model is constructed to satisfy the variety of virtual requests in the enterprise network, where a Gaussian distribution is used to reflect the computing capability of computers.

(3) The normalized method of computing and storage capabilities is proposed to determine whether physical nodes accommodate virtual nodes. The linear energy model of the CPU is improved to identify the actual utilization of computer nodes in the enterprise network.

(4) A model is designed to prove that EO-VNE is NP-complete. The EOLC algorithm is presented to optimize the energy consumption of the enterprise network.

The rest of this paper is briefly described as follows. The related work is summarized in Section II. EO-VNE model and EOLC are introduced in Section III. The performance evaluation of EOLC is shown in Section IV. The conclusion is presented in Section V.

II. RELATED WORK

VNE is a key issue in the field of network virtualization and has received extensive attention from researchers. For the review of previous studies, the problems of VNE mainly focus on virtual machine assignment and benefit optimization. We briefly review the representative work.

Chowdhury *et al.* [6] proposed a deterministic VNE embedding algorithm (DViNELB) that leveraged better coordination between the node and link phases. Dehury and Sahoo [7] addressed the dynamic resource demands in the VNE process. The purpose was to use maximum physical resources obtained from the local and global fitness value of virtual machines. In [8], an embedding model was built to address the resource allocation of the VNE problem, considering the computing, network and storage resources. In addition to virtual machine assignment, the energy consumption related to benefit optimization has attracted more researchers in recent years.

In [9], the base station virtual network protocol with firefly (BSVF) was presented to decrease the network energy consumption. BSVF was a meta-heuristic firefly algorithm that only shared information with neighboring nodes. The algorithm of energy-aware VNE with dynamic demands (EAD) was addressed in [10]. EAD modeled the dynamic demands as a Gaussian distribution and then designed a heuristic algorithm to minimize the energy consumption. In [11], energy-QoS balance in the cloud data center was achieved through the introduction of virtual machine integration technology. Zhang [12] presented a new node ranking scheme and differentiated pricing strategy to evaluate the node and link weights during the VNE process, which achieved a better trade-off between the energy consumption and network load balance. A hybrid algorithm was studied in [13]. This algorithm combined the greedy randomized adaptive search procedure and the reduced variable neighborhood search method to obtain the better energy efficiency in VNE. Davalos *et al.* [14] modeled the VNE problem as a multiobjective mixed integer linear problem (MILP) to maximize the benefits and minimize the costs. Fan *et al.* [15] proposed a multi-sleep mode scheduling scheme for servers. Different strategies were designed to determine which servers or when should sleep to save energy consumption. However, these investigations are designed for the cloud and may not be valid in the enterprise network. In addition, they did not consider the location constraint, which is very important for many real VNE applications.

To study the problem of the VNE with location constraint (LC), a multi-commodity flow problem was introduced, and MILP was designed to coordinate node and link mapping in [6], but it did not provide a good solution. In [16], a resource management approach was proposed to choose the locations, types and cloud instances, which met the performance requirements and reduced analysis costs. The genetic-based approach appeared in [17], which selected and leased virtual machines with lower costs and latency for users based on the location of service. A mechanism was proposed in [18] to detect the user's location and select the adequate cloud resources. A location constraint model based on a bisection graph (GLC) was proposed in [19], and this model was solved with the method of maximal clique under the condition that the substrate network had sufficient resources. In [20], the energy-aware and location-constrained virtual network embedding algorithm was proposed to decrease the energy consumption in the enterprise network. This algorithm only considered CPU and bandwidth requirements and could partially meet the enterprise demands.

III. ENERGY OPTIMIZATION AND LOCATION CONSTRAINT VNE

A. THE DEFINITIONS IN EO-VNE

The capabilities of computers in VNE can be divided into node capability and link capability. Hence, the complex number theory is introduced to represent the two attributes.

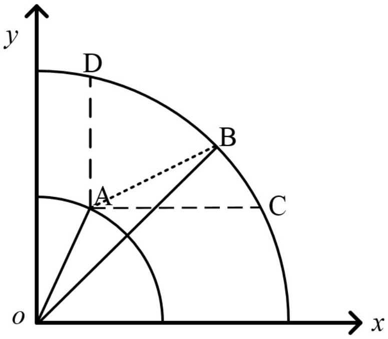

FIGURE 1. The condition of an acceptable virtual node.

Definition 1 (Network): A network is denoted as G, which is composed of a substrate network G_s and a virtual network G_v.

Definition 2 (Node): A complex number is defined to describe the capabilities of physical or virtual nodes. The real part value represents the node capability. The imaginary part value represents the link capability.

Definition 3 (Type of Virtual Nodes): A requirement type of virtual nodes is represented as a parameter whose value is equal to $arctg\left(b/c\right)$, where b is the imaginary part and c is the real part.

Definition 4 (Acceptable Virtual Node): When a physical node accommodates a virtual node, this virtual node is defined as an acceptable virtual node if it satisfies the condition that the slope K is greater than or equal to 0.

To normalize values of the real and the imaginary parts simultaneously in Definition 2, an approach is presented and demonstrated from two aspects. The value of the real part is computed by the quotient, which is defined as the node capability divided by the standard node capability. Additionally, the value of the imaginary part is computed in the same way, that is, its value is equal to link capability divided by the standard link capability. Note that the complex number of standard node is not a constant value. It should be set in advance according to the performance of computers in the enterprise network. However, it will not affect the performance of EOLC, regardless of the value of the standard node.

A detailed explanation of Definition 4 is shown in Figure 1. OA denotes the capabilities of a virtual node, and OB denotes the capabilities of a physical node. AC and AD are parallel to the x-axis and the y-axis, respectively. If the physical node can accommodate the virtual node, the location of point B should be between C and D, i.e., the slope K of AB is greater than or equal to 0.

B. EO-VNE NETWORK

The substrate network is modeled as an undirected weighted graph $G_s\left(V_s, E_s\right)$, where V_s denotes a set of physical nodes and E_s denotes a set of physical links.

The physical node is denoted as v_s^i, and the link is denoted as l_s^i. The link value is correlated to the starting node of this link, and the value varies over time.

Similar to G_s, the virtual network is $G_v\left(V_v, E_v, LC_v\right)$. The difference is that LC_v is introduced to represent the location constraint. The virtual node v_v^i can choose preferred physical nodes as the candidate set, which is denoted as $\Psi_{v_v^i}^{\rho} = \left\{v_s^k \in V_s \,\middle|\, \left\|v_s^k - v_v^i\right\| \leq \rho\right\}$, where ρ is a radius and $\left\|v_s^k - v_v^i\right\|$ is the distance between two nodes.

CPU requests, storage requests, and bandwidth requests are dynamic over time in the cloud platform. However, CPU resources are a major requirement for tasks in the enterprise network, neither storage resources nor link resources. Therefore, the variations of storage and bandwidth requests are more stable than the variations in the cloud. In this paper, the dynamic characteristics of storage and bandwidth requirements are ignored. According to the CPU data in [21], the CPU requirement obeys a Gaussian distribution with a mean value μ and a standard deviation σ.

An example of EO-VNE is shown in Figure 2. Figure 2(a) denotes the substrate network. Figure 2(b) denotes two virtual requests. Figure 2(c) denotes one of the solutions. To compute the available capabilities, the complex number of nodes is denoted as $R + Ii$. In (a), the provided resource value of A is $R_A + I_A i$. In (b), the requested resource value of a is $R_a + I_a i$, and the set of candidate nodes is $\{A\}$. Taking A as an example, a is embedded to A in the beginning of VNE, so the available resource value of A is computed as $(R_A - R_a) + (I_A - I_a)i$, which is $30 + 40i$. Then, b is embedded to C, so the available resource value of C is computed as $(R_C - R_b) + (I_C - I_b)i$, which is $30 + 35i$. However, the communication request between a and b is I_a, so the available resource value of C is recomputed as $(R_C - R_b) + (I_C - I_b - I_a)i$, which is $30 + 5i$. When c is embedded to D, the communication request between c and a is I_c; therefore, A is updated from $(R_A - R_a) + (I_A - I_a)i$ to $(R_A - R_a) + (I_A - I_a - I_c)i$, which is $30 + 20i$.

C. EO-VNE MODEL

Generally, the process of VNE is either node mapping priority or link mapping priority, but the embedding process in EO-VNE is a special scenario. The link capability can be integrated into node capability according to the complex number theory. Therefore, node embedding should be given more attention during the process of VNE.

The node capability is composed of computing and storage capabilities. To meet the actual rules of complex number, the values of both computing and storage capabilities should be normalized to a value. A node in G is denoted as i, and the computing and storage capabilities of i are defined as $com(i)$ and $stor(i)$, respectively. A weighting factor $\alpha \in [0, 1]$ is set to balance the computing and storage capabilities.

Node Capability Model: The node capability of i is marked as c^i, which can be calculated using (1).

$$c^i = \alpha * \frac{com\left(i\right)}{\sum\limits_{k \in V} com\left(k\right)} + (1 - \alpha) * \frac{stor\left(i\right)}{\sum\limits_{k \in V} stor\left(k\right)} \tag{1}$$

where k is the node in V.

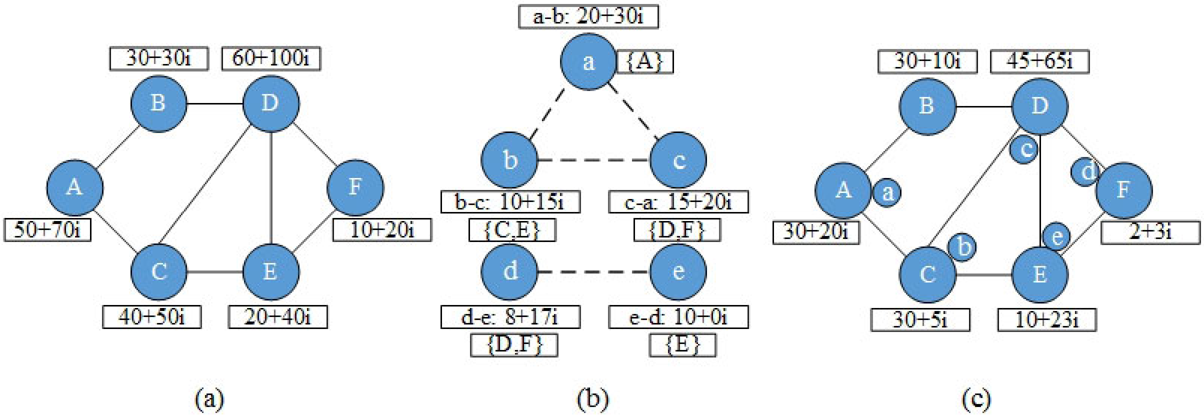

FIGURE 2. An example of EO-VNE.

Node Energy Model: To meet the actual situation of computer nodes in the enterprise network, the node energy linear model is improved using (2):

$$P = \begin{cases} P_b + (P_{max} - P_{min}) \cdot CPU & \textit{if the node is active} \\ 0 & \textit{otherwise} \end{cases} \tag{2}$$

where P_b is the baseline power, P_{max} and P_{min} are the full workload power and the empty workload power, respectively, and CPU is the current workload. Note that for the servers, P_b is equal to P_{min}. However, for the computer nodes, P_{min} is calculated using $P_b + (P_{max} - P_b) \cdot CPU'$, where CPU' is the official software workload. The computer may be a daily work node and an EO-VNE node at the same time. Hence, daily work may utilize software tools, which causes changes in the CPU and energy consumption. However, this part of energy is not calculated in the EO-VNE model.

In the process of VNE, there are two types of physical nodes, namely, the hosting node and the forwarding node. The energy consumption of each type of node is different. A node may also belong to two types at the same time, and in this situation, the type of node is the hosting node. Hence, the energy consumption in the period of T can be computed as:

$$E_1(T) = \int_0^T \Bigg[P_b (H(t) + F(t)) + (P_{max} - P_{min}) \sum_{i=1}^{V_v(t)} CPU_i(t) \Bigg] dt \tag{3}$$

In (3), $H(t)$ is the hosting node number at time t and $F(t)$ is the forwarding node number, $V_v(t)$ is the virtual node number, and $CPU_i(t)$ is the CPU requirement of virtual node i.

Link Energy Model: As shown in [10], the energy of the unit link is usually set as a fixed value P_{fix}. Thus, the link energy can be computed as:

$$E_2(T) = P_{fix} \cdot \int_0^T \sum_{i=1}^{V_v(t)} \sum_{v_v^{ij} \in \phi(v_v^i)} \sum_{v_s \in f\left(v_v^i \to v_v^j\right)} |v_s| \cdot b(t)\, dt \tag{4}$$

The link of two virtual nodes is $v_v^{ij} = \left\{ v_v^j \middle| v_v^j = \phi\left(v_v^i\right) \right\}$. In this link, v_v^i is the starting node and v_v^j is the ending node. When v_v^i sends data to v_v^j ($v_v^i \to v_v^j$), all the forwarding virtual nodes on this communication link can form a set $\phi\left(v_v^i\right)$. $f\left(v_v^i \to v_v^j\right)$ denotes the set of physical nodes that are located in the link between v_v^i and v_v^j. $|v_s|$ denotes the hop-count corresponding to $v_v^i \to v_v^j$. $b(t)$ denotes the link requirement satisfying $v_v^i \to v_v^j$ at time t.

The goal of EO-VNE is to obtain a solution scheme that makes the overall energy consumption less than or equal to a positive number EQ.

$$E = E_1 + E_2 \leq EQ \tag{5}$$

The EO-VNE solution is an embedding scheme from a virtual network to a physical network, and it needs to meet the following constraints.

(1) Node mapping constraint. A virtual node can only be embedded to one physical node in a VNE process. The embedding function is $g(\cdot)$.

$$v_s = g(v_v) \tag{6}$$

$$g\left(v_v^1\right) = g\left(v_v^2\right) \quad iff\ v_v^1 = v_v^2 \tag{7}$$

(2) Location constraint. The candidate set of a virtual node is denoted as follows.

$$v_s = \left\{ v_s^k \middle| v_s^k \in \Psi_{v_v^i}^{\rho}\ \&\ v_s^k = g\left(v_v^i\right) \right\} \tag{8}$$

(3) Node capability constraint. As shown in Definition 4, if a virtual node can be embedded to a physical node, then K is greater than or equal to 0. Note that the node that has the minimized link value located on the path of v_v^i and v_v^j should be found. The link value of this node is used to compute K.

$$K = \frac{min\left(b_s^{v_s}\right) - b_v^{v_v^{ij}}}{c_s^{v_s^k} - c_v^{v_v^i}} \geq 0, \quad if\ v_s^k = g\left(v_v^i\right)\ \&\ v_s \in f\left(v_v^i \to v_v^j\right) \tag{9}$$

D. EO-VNE COMPLEXITY

The bisection problem is NP-complete [19]. Specifically, given $G(V, E)$ and a nonnegative integer Z, whether there exist two subsets V_1 and V_2, satisfying $V = V_1 \cup V_2$, $V_1 \cap V_2 = \phi$, $|V_1| = \lceil |V|/2 \rceil$, $|V_2| = \lfloor |V|/2 \rfloor$, and the number of links between V_1 and V_2 is not more than Z.

Lemma 1: The EO-VNE problem is NP-complete.

Proof: First, the EO-VNE problem is a *NP* problem. When an embedding scheme is found, the determination of whether (6) - (9) are satisfied can be completed in polynomial time.

Then, we prove that EO-VNE has a desired solution if and only if there is a desired bisection graph. As shown in the bisection graph $G(V, E)$, EO-VNE has $G_v(V_v, E_v)$ satisfying $V_v = V$ and $E_v = E$. The node comes directly from the virtual network, which is denoted by $V_v^1 = \left\{\alpha_v^i\right\}(i \in [1, |V_v|])$, where node α_v^i comes from G_v. The situation of link E_v^1 is similar to V_v^1. The subgraph with V_v^1 and E_v^1 is denoted as $\alpha - island$, and the other set is $V_v^2 = \left\{\beta_v^k\right\}$, where $k \in [1, |V_v|]$. The nodes in this set are extra nodes. The links are $E_v^2 = \phi$. V_v^2 and E_v^2 form a subgraph, which is $\beta - island$. β_v^i and α_v^i are one-to-one correspondences.

In the substrate network, two complete subgraphs $\alpha' - island$ and $\beta' - island$ can be constructed as follows.

$$\begin{aligned} V_s^1 &= \bigcup \left\{\alpha_s^i\right\}, \\ E_s^1 &= \left\{\left(\alpha_s^i, \alpha_s^j\right) \in V_s^1 \times V_s^1, i \neq j, i, j \in \left[1, \left|V_s^1\right|\right]\right\} \end{aligned} \tag{10}$$

$$\begin{aligned} V_s^2 &= \bigcup \left\{\beta_s^k\right\}, \\ E_s^2 &= \left\{\left(\beta_s^k, \beta_s^h\right) \in V_s^2 \times V_s^2, k \neq h, k, h \in \left[1, \left|V_s^2\right|\right]\right\} \end{aligned} \tag{11}$$

The bridge needs to be built to connect $\alpha' - island$ and $\beta' - island$. Only two nodes chosen from $\alpha' - island$ and $\beta' - island$ are denoted as in-bridge o_s^{in} and out-bridge o_s^{out}, respectively. These two nodes are the initiation and termination of the bridge. The nodes in the $\alpha' - island$ are all connected to o_s^{in}. The nodes in the $\beta - island$ are all connected to o_s^{out}. Two types of paths can be built in the bridge: the direct connection and the Δ-hops connection. The detailed descriptions are as follows.

$$V_s' = \bigcup \left\{v_s^i\right\} \bigcup \left\{o_s^{in}, o_s^{out}\right\} \; i \in [1, \Delta + 1] \tag{12}$$

where v_s^i is the node in the Δ-hops line topology.

$$E_s^{1'} = \left\{\left(\alpha_s^i, o_s^{in}\right) \middle| \alpha_s^i \in V_s^1\right\} \bigcup \left\{\left(\beta_s^i, o_s^{out}\right) \middle| \beta_s^i \in V_s^2\right\} \tag{13}$$

where $E_s^{1'}$ is the set of links in the Δ-hops line topology.

$$E_s^{2'} = \left\{\left(o_s^{in}, o_s^{out}\right)\right\} \tag{14}$$

where $E_s^{2'}$ is the first type of path.

$$\begin{aligned} E_s^{3'} = \left\{\left(o_s^{in}, v_s^i\right)\right\} \bigcup \left(\bigcup \left\{\left(v_s^i, v_s^k\right) \middle\| v_s^i \rightarrow v_s^k \right| = \Delta \right\}\right) \\ \bigcup \left\{\left(v_s^k, o_s^{out}\right)\right\} \end{aligned} \tag{15}$$

where $E_s^{3'}$ is the second type of path. $\left|v_s^i \rightarrow v_s^k\right|$ is the number of forwarding nodes located in the communication path.

On this basis, the location constraints are described as $\Psi_{\alpha_v^i}^{\rho} = \left\{\alpha_s^i, \beta_s^i\right\}$, $\Psi_{\beta_v^i}^{\rho} = V_s^1$ *if* $i \leq \lceil |V|/2 \rceil$ and $\Psi_{\beta_v^i}^{\rho} = V_s^2$ *if* $i > \lfloor |V|/2 \rfloor$. When the physical nodes have a unit energy of both node capability and link capability, *EQ* can be computed using the following formula.

$$EQ = 2\left(|V| + \lfloor |V|/2 \rfloor^2 + \lceil |V|/2 \rceil^2\right) + |E| + |V|^2 \tag{16}$$

All the steps mentioned above can be done in polynomial time.

Then, the node mapping function that embeds virtual nodes to physical nodes is constructed as follows.

$$\Theta\left(\beta_v^i\right) = \begin{cases} \alpha_s^{\bar{i}} & \alpha_v^{\bar{i}} \in V_1 \& i \leq \lceil |V|/2 \rceil \& \\ & \alpha_v^i \bigcap \alpha_v^{\bar{i}} = \phi \& \alpha_v^i \bigcup \alpha_v^{\bar{i}} = V_1 \\ \beta_s^{\bar{i}} & \alpha_v^{\bar{i}} \in V_2 \& i > \lfloor |V|/2 \rfloor \& \\ & \beta_v^i \bigcap \beta_v^{\bar{i}} = \phi \& \beta_v^i \bigcup \beta_v^{\bar{i}} = V_2 \end{cases} \tag{17}$$

When the node mapping function is determined, the link mapping function is obvious. Based on the above steps, EO-VNE can be converted to a bisection graph. Therefore, EO-VNE is *NP*-complete. □

E. THE SOLUTION OF EO-VNE

From Lemma 1, the solution of EO-VNE cannot be found in polynomial time, so the energy-optimized virtual network embedding with the location constraint algorithm (EOLC) is proposed to find the near-optimal solution.

In the enterprise network, two different virtual nodes cannot be mapped to one physical node in a VNE process, which satisfies the requirement of a bisection graph. At the same time, physical nodes that provide better QoS are usually selected by more than one virtual node, such that $V_1 \bigcap V_2 \neq \phi$. Therefore, the predealing process is designed to satisfy $V_1 \bigcap V_2 = \phi$. Three methods are presented. (1) The virtual node with the smallest candidate set is found; then the candidate nodes are randomly chosen and removed from the candidate set of remaining virtual nodes. (2) The number of each candidate node is computed, and the smallest one is determined. The virtual node with this candidate node is then randomly chosen. This candidate node is removed from the candidate set of remaining virtual nodes. (3) As in (2), the candidate node with the largest number is chosen.

After the predealing process, a matching graph (MG) is constructed to unify the bisection graph, node capability and link capability. Two construction rules are obeyed. (1) A node in MG is a candidate physical link. For example, two virtual nodes a and b have their candidates $\{c\}$ and $\{d, e\}$. If a wants to communicate to b, two nodes are created in the matching graph. The first node is d when d is the neighbor of c in the substrate network, i.e., (c, d). The other node is f when e is not the neighbor of c but f is the neighbor of both c and e, i.e., (c, f), (f, e). (2) When two links have a common physical node, the link in MG is drawn. In other words, the two links can accommodate two non-adjacent candidate links, or they are candidate paths of two adjacent links. The detailed predealing process and MG construction are shown in Algorithm 1.

Similar to [22], the solution of EOLC is equivalent to finding the maximum set of MG. This set should satisfy covering the links of VRs and meet the constraints in (6) - (9). Hence, heuristic Algorithm 2 is presented to obtain the near-optimal maximum set.

IV. SIMULATION

A. SIMULATION SETUP

Because the GT-ITM tool [23] can be widely used in generating the network topology, we adopt it to generate and simulate the enterprise network environment. The generated topology can simulate the real network and finally produce realistic simulation results. The experiments run on the substrate network with 50 computers and 4 servers. Similar to the work [10], [19], the network grid is set as 100 × 100. The computers are randomly mapped into the grids, but the servers are in the same grid. This setting is suited for an actual enterprise network, where the computers in various offices are usually located in different rooms, but the servers are in the same room. The number of virtual network requirements is randomly selected from 1 to 10. The probability of any two virtual node connections is 0.5. v_v^i selects some physical nodes located in a circle, and finally, these selected physical nodes form a candidate set, where the number of nodes it contains is any value from 1 to 10. The center of the circle is v_v^i, and the radius r is between 15 and 30. The virtual requirement arrival rates follow the Poisson process in which the average arrival rate is 5 per 10 minutes and the holding time is the exponential distribution of 500 minutes. Parameter α is 0.8. The CPU requirements of virtual nodes obey a Gaussian distribution with $N(\mu, \sigma)$, $\mu \in [0, 20]$ and $\sigma \in [0, 10]$. The value of parameter P_b is 165 W and $P_{max} - P_{min}$ is 15 W/CPU unit [10].

B. PERFORMANCE VALUATION

The MILP algorithm [6] is adopted as a benchmark for evaluating the solution of EOLC. This algorithm was adjusted to fit the scenario in this paper, including two modifications. The first one is to extend node capability to a combination of computing and storage capabilities. The second one is to add the judgement of the node location. A detail description of comparison algorithms is shown in Tab.1. Then, the optimal solution E^* and the solution of EOLC E are obtained. A parameter ω is introduced to denote the distance between E^* and E. This parameter is calculated by $\omega = (E - E^*)\big/E$. The other parameter named blocking probability is set to denote the probability of unserveable VRs. It is calculated using $\varpi = \lim_{t \to T} \frac{|\Gamma_{blocked}(t)|}{|\Gamma_{blocked}(t)| + |\Gamma_{accepted}(t)|}$, t and T are the current time and the deadline time, respectively. The servers are usually powered within 24 hours, so $T \to +\infty$. $\Gamma_{blocked}(t)$ is the set of blocked VRs and $\Gamma_{accepted}(t)$ is the accepted VRs.

Algorithm 1 The Predealing Process and Matching Graph Construction

```
Input: The complex numbers of both physical and virtual nodes and the candidate sets of virtual nodes.
Output: MG.
Calculate the sum of computing and storage capabilities of virtual nodes with $\sum_{v_v \in V_v} com(v_v)$ and $\sum_{v_v \in V_v} stor(v_v)$, respectively.
Calculate the sum of computing capability and storage capability of physical nodes with $\sum_{v_s \in V_s} com(v_s)$ and $\sum_{v_s \in V_s} stor(v_s)$, respectively.
Calculate the value of adjusting factor as: $\zeta = \alpha \frac{\sum_{v_v \in V_v} com(v_v)}{\sum_{v_s \in V_s} com(v_s)} + (1 - \alpha) \frac{\sum_{v_v \in V_v} stor(v_v)}{\sum_{v_s \in V_s} stor(v_s)}$.
for (each $v_v \in V_v$) do
    Calculate the virtual node capability $c^{v_v}$ with (1), then adjust the value of $c^{v_v}$ as $c^{v_v} = c^{v_v} * \zeta$.
    for (each $v_s \in \Psi_{v_v}^{\rho}$) do
        Calculate the physical node capability with (1).
        if ($K < 0$) then
            Remove $v_s$ from $\Psi_{v_v}^{\rho}$.
        end if
    end for
end for
Utilize the predealing process method to form a new candidate set $v_s \in \Psi_{v_v}^{\rho}$ corresponding to each $v_v$.
for (each connection between $v_v^i$ and $v_v^j$) do
    for (each $v_s^i \in \Psi_{v_v^i}^{\rho}$) do
        for (each $v_s^j \in \Psi_{v_v^j}^{\rho}$) do
            if ($v_s^i$ is directly connected to $v_s^j$) then
                Add a new node in MG and mark it as "$v_s^i$-$v_s^j$".
            else
                Find a path between $v_s^i$ and $v_s^j$ with the shortest path algorithm, and mark with "$v_s^i$-...-$v_s^j$".
            end if
        end for
    end for
end for
Mark the nodes to represent the same candidate path of $v_s^i$ and $v_s^j$ in one group.
for (each $v_{MG}^m \in$ MG) do
    for (each $v_{MG}^n \in$ MG & $m \neq n$ & $v_{MG}^m$ and $v_{MG}^n$ are not in the same group) do
        if (two adjacent candidate paths accommodate the same common node "$v_s^i$" | two adjacent candidate paths accommodate two virtual links) then
            Create a link to connect $v_{MG}^m$ and $v_{MG}^n$.
        end if
    end for
end for
Output MG.
```

Figure 3 is the comparison of ω with DVINELB, GLC and EOLC. Each data point is 5 VRs. The disparity is small when the algorithm introduces load balancing because load

TABLE 1. Comparison Algorithms.

Algorithm	Node Embedding	Link Embedding	Targets
DVINELB	Solving MCF-based MILP with load balance, and using deterministic and randomized rounding approach	Solving MCF with load balance	Increasing the acceptance ratio and the revenue
GLC	Solving the minimum cost maximum clique	Using Greedy and shortest substrate path	Reducing the time complexity and increasing the performance of VNE
EAD	Utilizing the strike balance method to guarantee high revenue	Using shortest substrate path	Leveraging the dynamic characteristic of VNE and using the energy model to decrease the energy consumption and increase the revenue of physical networks
BSVF	–	–	Using the meta-heuristic firefly algorithm with load balance to decrease the energy of base stations
EOLC	Using the complex number theory to integrate the node and link capabilities and utilizing the near-optimal maximum set algorithm	Lightweight link embedding	Decreasing the energy consumption of physical networks

Algorithm 2 The Near-Optimal Maximum set Algorithm

```
Input: $G_s$, $G_v$, MG
Output: The maximum set $M$
if ($v_v^i$ is the neighbor of $v_v^j$ & $v_s^i = g\left(v_v^i\right)$ & $v_s^j = g\left(v_v^j\right)$ & (“$v_s^i$ -...-$v_s^j$” || $v_s^i$“ - $v_s^j$”) is the mark of a node $\Lambda$ in MG) then
    Compute the bandwidth requirement $b_v^{v_i,v_j}$.
end if
Create a set $DS$ based on $b_v^{v_i,v_j}$ sorted in descending order.

for (each $b_v^{v_i,v_j} \in DS$) do
    if ($min\left(b_s^{v_s^k}\right) \geq b_v^{v_i,v_j}$, $v_s^k \in \left\{v_s^i, \cdots, v_s^j\right\}$ & $c_s^{v_i} < c_v^{v_i}$ & $c_s^{v_j} < c_v^{v_j}$ ) then
        $M' \leftarrow \Lambda$.
    else
        Return “Finding the maximum set failed!”.
    end if
    Select the node in $M'$ that has the minimum tops, then add this node to $M$.
    Update the real and imaginary parts of the physical nodes.
end for
Return $M$;
```

balancing causes the algorithm to choose higher resource consumption to make the remaining resources more adequate. EOLC and GLC can provide similar optimization disparity, but EOLC requires less execution time than GLC because the introduction of complex number theory makes EOLC focuses on node mapping, i.e., the link embedding only occurs when a new node is embedded.

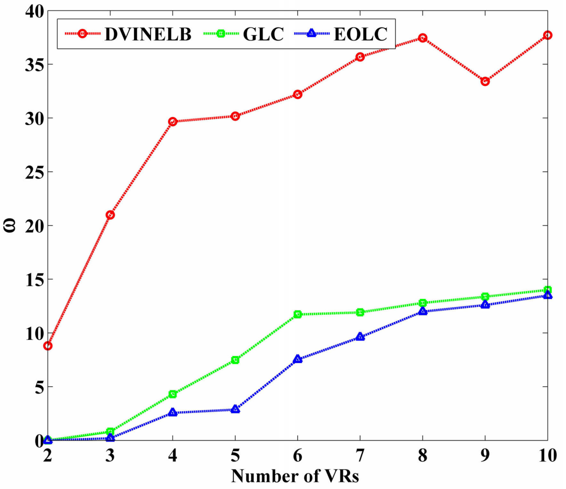

FIGURE 3. The disparity of optimization solution ω.

Figure 4 demonstrates the evaluation of Algorithm 1. Each data point is 5 VRs. As the number of VRs and the radius r increase, the probability of overlapping candidate sets with different VRs will increase, which causes ω to become larger. Although when the number of VRs is 10 and r is 30, ω is still less than 15%, which is relatively smaller compared to the results of other heuristic algorithms. Figure 5 shows the number of solutions found using different heuristic algorithms. Each data point is 5 VRs. In the simulations, the maximum number of solutions is 30. An interesting observation is that the number of solutions found using EOLC is relatively smaller than the number found using DVINELB because in DVINELB, the multipath mapping is allowed, while EOLC is based on mapping on nodes and it only allows the single-path mapping.

The comparison results of ϖ are shown in Figure 6. The blocking probability values of EOLC are between those of

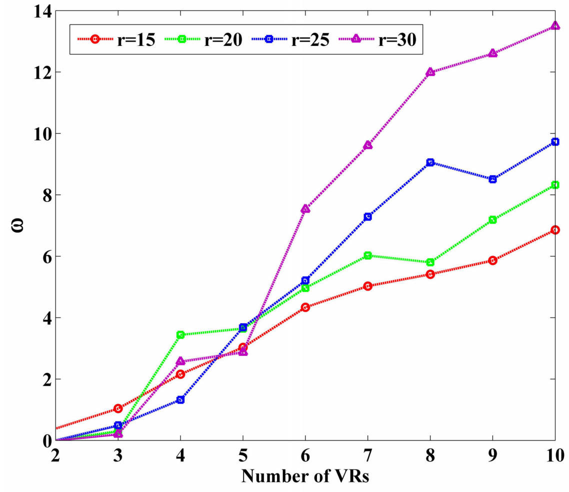

FIGURE 4. The disparity of optimization solution with *r*.

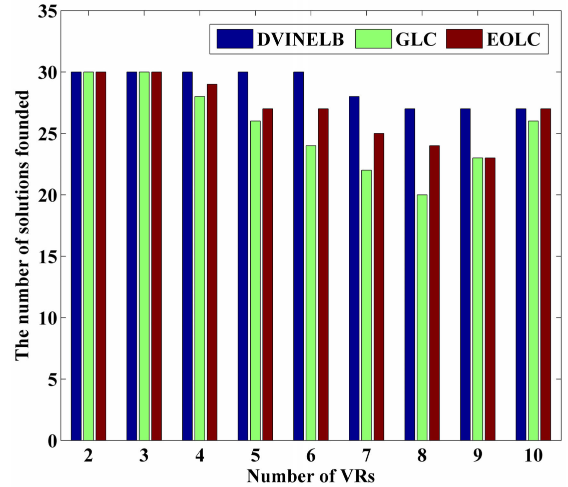

FIGURE 5. The number of solutions found using three algorithms.

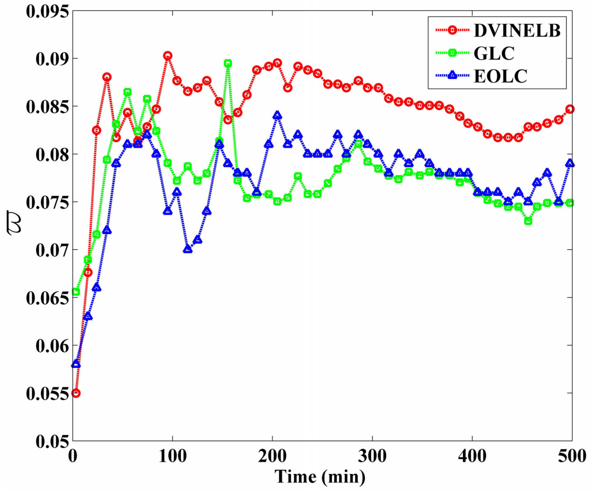

FIGURE 6. The obtained ϖ using three algorithms.

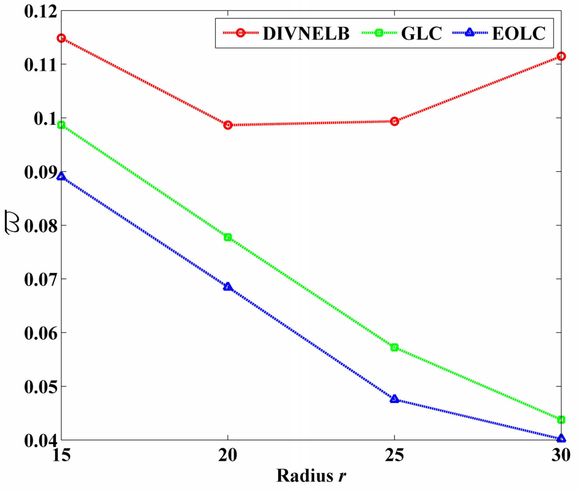

FIGURE 7. Blocking probability values of different radii.

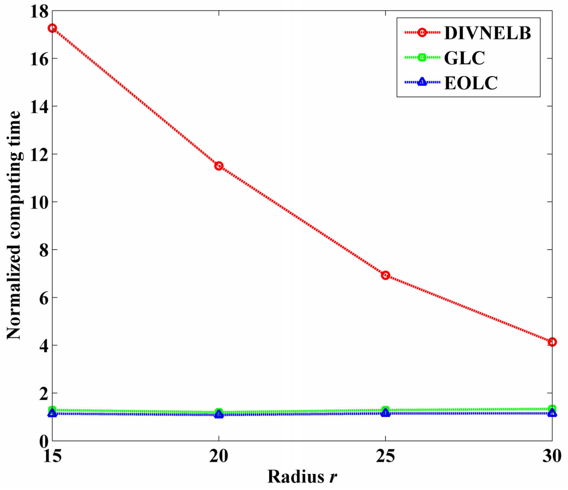

FIGURE 8. The normalized computing time of different radii.

DVINELB and GLC because GLC has adequate resources. These resources guarantee obtaining the optimal solution for a long time. However, EOLC is used for the enterprise network, and the remaining resources are not enough to obtain the best optimal solution in all time. Therefore, EOLC with a parameter ϖ performs worse than GLC.

Figure 7 shows the performance of EOLC. In Algorithm 1, the candidate sets will overlap among different VRs, which makes EOLC perform worse. Hence, the blocking probability values with different r are evaluated. From this figure, the EOLC algorithm can achieve lower blocking probability values than other algorithms. Note that the number of candidate sets increases when r increases, causing the solution space to increase, which is the reason that the EOLC algorithm performs better. However, the performance is not always good, especially when the node mapping encounters the bottleneck.

Figure 8 shows the time complexity of EOLC. The computing time of EOLC is set as a benchmark, and this benchmark is used to convert the time of other algorithms into the normalized computing time. The simulation environment is composed of a server with 3.3GHz Intel Xeon

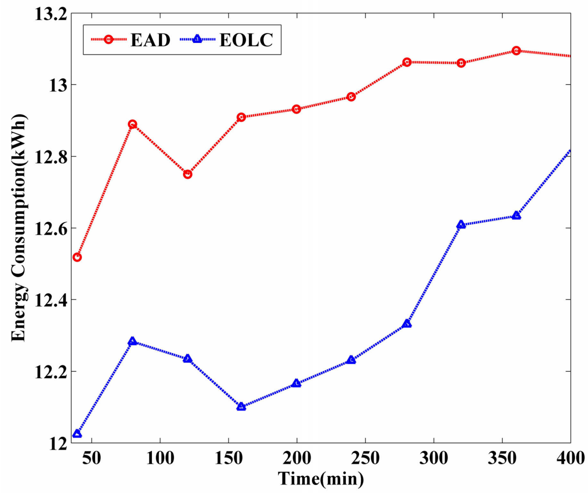

FIGURE 9. The average energy consumption of VRs with small-sized scale.

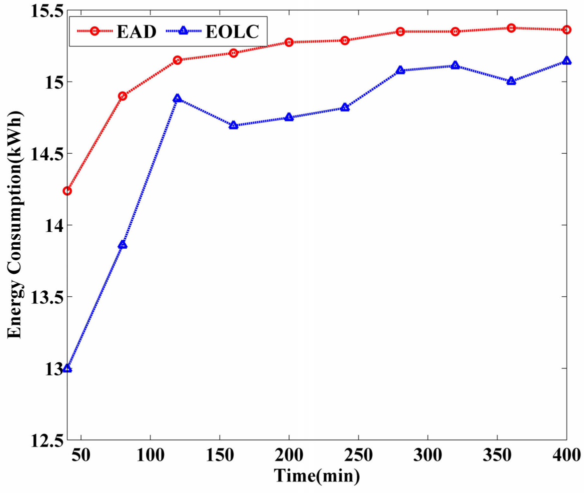

FIGURE 10. The average energy consumption of VRs with medium-sized scale.

E3-1225 v5 CPU and 16G RAM. The results demonstrate that the EOLC algorithm requires considerably less computing time, especially when *r* is small because the DVINELB algorithm must solve the linear programming problem twice, which may require considerable computing time. Benefiting from Algorithm 1, EOLC can find the solution space with fewer iterations than GLC. It is also found that the computing time is less affected by the network size. The parameter *r* is the key factor which can affect the complexity of EOLC. When *r* is large, the candidate set of a virtual request is large. However, after predealing process, there may not be many vertexes in constructed matching graph, so the computing time of finding solutions will not be affected. Noting that, when the size of matching graph is large, it may get the solutions easily, so as to decrease the computing time.

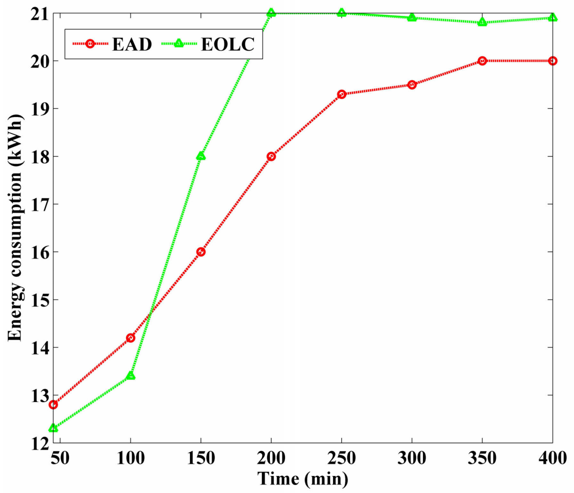

FIGURE 11. The average energy consumption of VRs with large-sized scale.

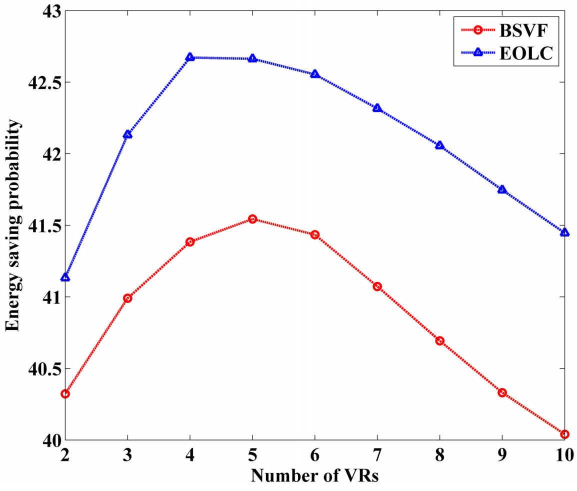

FIGURE 12. Comparison of energy saving probability using BSVF and EOLC.

C. ENERGY CONSUMPTION VALUATION

Figure 9 and Figure 10 evaluate energy consumption using EOLC and EAD. In Figure 9, when VRs with small-sized scale arrive, EOLC has less energy consumption than the EAD algorithm. In addition, the remaining resources are abundant in the enterprise network when the size of the VR scale is small. Therefore the physical nodes can be relatively concentrated, and the number of forwarding nodes can be reduced. However, as shown in Figure 10, when VRs with medium-sized scale arrive (the VRs arrival rate is adjusted to 10 per 10 minutes), the curves of EAD and EOLC become closer because as the number of VRs increases, the remaining resources of current physical nodes decreases. This makes it more difficult to calculate the best solution of EOLC. Therefore, more physical nodes are needed to join the substrate network, resulting in extra energy consumption. Especially when the size of the VR scale is large, the energy consumption can exceed that of the EAD. As shown in Figure 11, the energy consumption of EAD is less than that of EOLC when the scale size of VRs is large (the VR arrival rate is adjusted to 15 per 10 minutes) because EAD considers the subsequent bandwidth requirements and performs link mapping in advance, thereby improving the use efficiency of the substrate link resources. However, the link embedding process accompanies the node embedding process, so it is difficult to consider the bandwidth requirements in advance.

An additional evaluation of energy saving probability compared to the BSVF algorithm is performed to show the performance of EOLC, as shown in Figure 12. Each data point is 10 VRs. Because the nodes in EOLC are always powered on in the enterprise network, however, when the nodes in BSVF are useless, they can be shut down or put in sleep mode. Therefore, the algorithms EOLC and BSVF only consider the VR mapping process and executing process. Additionally, the EOLC algorithm has a much higher energy saving probability in view of reducing the time of VNE and avoiding frequent shifting between nodes.

V. CONCLUSION

Intergrading the available resources in the enterprise network is an effective approach to improve the data processing capability of the enterprise. In this approach, the virtual technology can be utilized and the energy consumption must be considered. So, in this paper, the energy-optimized VNE problem with a location constraint is proposed and solved. First, the complex number theory is introduced to unify the computing, storage and link capabilities into node capability. The EO-VNE network is built and the capabilities of nodes are computed with the complex number theory. Then, the energy optimization problem is modeled as EO-VNE. In this model, the energy consumption of nodes and links is calculated with the location constraint. Third, a method for converting EO-VNE into a bisection graph is proposed, and EO-VNE is shown to be NP-complete. Fourth, to obtain the optimal solution of EO-VNE, a heuristic algorithm EOLC is designed. Finally, the performance of EOLC is evaluated in detail with the disparity of optimization solution, the number of solutions, the blocking probability, the computing time, and the energy consumption. The evaluation results indicate that the performance of EOLC is better and the energy consumption is lower than EAD and BSVF.

In the future, the heuristic algorithm will be improved to fit long-time virtual network embedding of enterprise networks. And the restriction on bandwidth will be considered suitable for multi-region enterprise networks.

ACKNOWLEDGMENT

This article was presented in part at the International Conference on Service Oriented Computing Workshop, Hangzhou, China, in 2018.

REFERENCES

[1] N. Ogino, T. Kitahara, S. Arakawa, and M. Murata, "Virtual network embedding with multiple priority classes sharing substrate resources," *Comput. Netw.*, vol. 112, pp. 52–66, Jan. 2017.

[2] G. Chochlidakis and V. Friderikos, "Mobility aware virtual network embedding," *IEEE Trans. Mobile Comput.*, vol. 16, no. 5, pp. 1343–1356, May 2017.

[3] Z. Cao, J. Lin, C. Wan, Y. Song, Y. Zhang, and X. Wang, "Optimal cloud computing resource allocation for demand side management," *IEEE Trans. Smart Grid*, vol. 8, no. 4, pp. 1943–1955, Jul. 2017.

[4] A. Yousafzai, A. Gani, R. M. Noor, M. Sookhak, H. Talebian, M. Shiraz, and M. K. Khan, "Cloud resource allocation schemes: Review, taxonomy, and opportunities," *Knowl. Inf. Syst.*, vol. 50, no. 2, pp. 347–381, Feb. 2017.

[5] A. Fischer, J. F. Botero, M. T. Beck, H. de Meer, and X. Hesselbach, "Virtual network embedding: A survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 4, pp. 1888–1906, 4th Quart., 2013.

[6] M. Chowdhury, M. R. Rahman, and R. Boutaba, "ViNEYard: Virtual network embedding algorithms with coordinated node and link mapping," *IEEE/ACM Trans. Netw.*, vol. 20, no. 1, pp. 206–219, Feb. 2012.

[7] C. K. Dehury and P. K. Sahoo, "DYVINE: Fitness-based dynamic virtual network embedding in cloud computing," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 1029–1045, May 2019.

[8] P. Zhang, H. Yao, and Y. Liu, "Virtual network embedding based on computing, network, and storage resource constraints," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3298–3304, Oct. 2018.

[9] H. Zhang, J. Cai, and X. Li, "Self-organized virtual small networking for energy saving and load balancing in cellular networks," in *Proc. IEEE Int. Conf. Commun. Workshop (ICCW)*, Jun. 2015, pp. 2874–2879.

[10] Z. Zhang, S. Su, J. Zhang, K. Shuang, and P. Xu, "Energy aware virtual network embedding with dynamic demands," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2015, pp. 5386–5391.

[11] S. Y. Zahedi Fard, M. R. Ahmadi, and S. Adabi, "A dynamic VM consolidation technique for QoS and energy consumption in cloud environment," *J. Supercomput.*, vol. 73, no. 10, pp. 4347–4368, Oct. 2017.

[12] P. Zhang, "Incorporating energy and load balance into virtual network embedding process," *Comput. Commun.*, vol. 129, pp. 80–88, Sep. 2018.

[13] S. M. A. Araujo, F. S. H. de Souza, and G. R. Mateus, "Virtual network embedding in multi-domain environments with energy efficiency concepts," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Jan. 2018, pp. 205–210, doi: 10.1109/ICOIN.2018.8343111.

[14] C. Aceval, E. Dávalos, V. Franco, and B. Barán, "A multi-objective approach for VNE problems using multiple ILP formulations," *CLEI Electron. J.*, vol. 19, pp. 1–15, Aug. 2016.

[15] L. Fan, C. Gu, L. Qiao, W. Wu, and H. Huang, "GreenSleep: A multi-sleep modes based scheduling of servers for cloud data center," in *Proc. 3rd Int. Conf. Big Data Comput. Commun. (BIGCOM)*, Aug. 2017, pp. 368–375, doi: 10.1109/BIGCOM.2017.16.

[16] A. Mohan, A. S. Kaseb, Y.-H. Lu, and T. J. Hacker, "Location based cloud resource management for analyzing real-time videos from globally distributed network cameras," in *Proc. IEEE Int. Conf. Cloud Comput. Technol. Sci. (CloudCom)*, Dec. 2016, pp. 176–183, doi: 10.1109/CloudCom.2016.0040.

[17] T. Shi, H. Ma, and G. Chen, "A genetic-based approach to location-aware cloud service brokering in multi-cloud environment," in *Proc. IEEE Int. Conf. Services Comput. (SCC)*, Jul. 2019, pp. 146–153, doi: 10.1109/SCC.2019.00034.

[18] H. Bangui, S. Rakrak, and S. Raghay, "Selecting location-based services in mobile cloud computing," in *Proc. 11th Int. Conf. Intell. Syst., Appl. (SITA)*, Oct. 2016, pp. 1–5, doi: 10.1109/SITA.2016.7772296.

[19] L. Gong, H. Jiang, Y. Wang, and Z. Zhu, "Novel location-constrained virtual network embedding LC-VNE algorithms towards integrated node and link mapping," *IEEE/ACM Trans. Netw.*, vol. 24, no. 6, pp. 3648–3661, Dec. 2016.

[20] X. Cong, L.-L. Zi, and K. Shuang, "Energy-aware and location-constrained virtual network embedding in enterprise network," in *Proc. Int. Conf. Service-Oriented Comput. Workshops*, Apr. 2019, pp. 41–52.

[21] M. Wang, X. Meng, and L. Zhang, "Consolidating virtual machines with dynamic bandwidth demand in data centers," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 71–75.

[22] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, CA, USA: Freeman, 1990.

[23] O. Heckmann, M. Piringer, J. Schmitt, and R. Steinmetz, "Generating realistic ISP-level network topologies," *IEEE Commun. Lett.*, vol. 7, no. 7, pp. 335–336, Jul. 2003.

XIN CONG received the Ph.D. degree in computer science from the Beijing University of Posts and Telecommunications. He is currently an Associate Professor with the School of Electronic and Information Engineering, Liaoning Technical University.

KAI SHUANG received the Ph.D. degree in computer science from the Beijing University of Posts and Telecommunications. He is currently a Professor with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications.

LINGLING ZI received the Ph.D. degree in computer science from the Beijing University of Posts and Telecommunications. She is currently an Associate Professor with the School of Electronic and Information Engineering, Liaoning Technical University.