IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 18, 2019, accepted April 1, 2019, date of publication April 12, 2019, date of current version April 24, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2910807

Markov Approximation Method for Optimal Service Orchestration in IoT Network

WENCHEN HE[1], SHAOYONG GUO[1], YUN LIANG[2], AND XUESONG QIU[1]
[1]State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
[2]Global Energy Interconnection Research Institute Co., Ltd., Beijing 102200, China

Corresponding author: Shaoyong Guo (syguo@bupt.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61702048, and in part by the Beijing University of Posts and Telecommunications Excellent Ph.D. Students' Foundation.

ABSTRACT With the evolution toward 5G, the Internet of Things (IoT) is expected to manage resource dynamically and provide customized services for different users in a cost-effective manner. Many technologies, such as edge computing and network function virtualization (NFV), have responded to these demands and been even more critical for system mobility, scalability, flexibility, and resource utilization. In this vein, the efficient provisioning method of the IoT services in distributed clouds based on the diverse quality of experience (QoE) requirements is highly needed. This paper focuses on service chain (SC) orchestration and studies the optimal placement of virtual network functions (VNFs) with multiple instances to minimize cost and delay, as well as guarantee network load balancing. We propose a multi-objective optimization problem model and then convert this combinatorial optimization problem to the one which can be solved by distributed methods based on the Markov approximation. At last, a VNF placement with multiple instances algorithm (VPMIA) based on Markov chain is designed to solve the above problem in a distributed manner. The simulation results show that the proposed algorithm can outperform the random placement algorithm and the single-path placement algorithm in cost saving by 22% and 31%, respectively, with a high SC acceptance rate. Besides, it can guarantee the QoE requirements and make the network load balanced with the different numbers of SCs.

INDEX TERMS Distributed algorithm, Internet of Things, load balancing, Markov approximation, multiple instances, service chain, VNF placement.

I. INTRODUCTION

The Internet of Things (IoT) is supposed to connect a mass of terminal devices (i.e., connected cars, mobile devices and sensors) and support services with various requirements in terms of ultra-short delay and ultra-high bandwidth. Network Function Virtualization (NFV) and edge computing have become the key concepts to flexible IoT service provisioning as well as Quality of Experience (QoE) requirements guarantee [1]–[3]. NFV decouples integrated structure of hardware and software so that virtual network functions (VNFs) are flexibly instantiated in Virtual Machines (VMs) as software components. The placement of VNFs banded with traffic routing are known as service chain (SC) [4], [5]. Meanwhile, the distributed edge clouds (i.e., virtualized server clusters or small data centers) are deployed in plenty of locations in network, and thus able to provide CPU, memory resources closer to users compared with centralized cloud [6].

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

Combined with edge clouds, mobile network provides Mobile Edge Computing (MEC) capacity within the range of User Equipment (UE), which not only guarantees the mobility and QoE requirements of users, but also reduces transmission consumption of computation-intensive or delay-sensitive services from UEs to a centralized cloud [7].

In mobile network, the crucial network functions, such as Packet Data Network Gateway (PGW) and Serving Gateway (SGW), are virtualized completely and instantiated in clouds. The SGW and PGW are connected to e-NodeB and transmit service traffic of end users to Internet. This traffic generally needs different additional VNFs, such as WAN acceleration (WA), intrusion detection system (IDS), deep packet inspection (DPI), traffic shaper (TS). VNFs (i.e., PGW-C, SGW-C) associated with control plane are suitable for centralized placement and cannot be distributed in network. Instead, VNFs (i.e., WA, IDS, DPI, TS) related to data plane or some applications are used to process great amounts of traffic, and should be distributed with the indefinite number of instances in mobile edge clouds [8]. Hence the customized placement and chaining of these VNFs in distributed edge clouds have more benefits for cost efficiency and load balancing.

Specifically, for one SC request, the VNF placement with maximum instances in mobile edge clouds can make service traffic be scheduled to adjacent VNFs, which can decrease forwarding cost as well as balance the network load. However, this solution requires huge consumption of server resources. Conversely, placing VNFs with minimum instances reduces the consumption of server resources, but is of no help to QoE guarantee and load balancing. Therefore, the optimal service orchestration involves the optimal number, placement of VNF instances and traffic routing, which has combinatorial and non-convex characteristics. On the other hand, the VNF placement with multiple instances has been proved to be a NP-hard problem. Traditional optimization algorithms in a centralized manner have limited computational efficiency but strict demand for computing capacity for a single machine, which makes it difficult to be applied in engineering practice.

In this paper, we focus on the instantiation and chaining of VNF instances, and allow one kind of VNF to have the reasonable number of instances to decrease the forwarding cost. The technical contributions of this paper are summarized as follows.

- Two load balancing factors are designed to maintain the network load balanced. Then, considering the cost-efficiency and QoE guarantee, we formulate the problem of VNF placement with multiple instances and traffic routing as a multi-objective optimization problem model;
- Since above combinatorial optimization problem is difficult to be solved by traditional centralized algorithms, we convert it to the problem which can be solved in a distributed manner according to Log-Sum-Exp approximation method;
- Further, a VNF placement with multiple instances algorithm (VPMIA) based on Markov chain is designed to solve above problem. Moreover, the effectiveness of proposed algorithm compared with two centralized algorithms is also evaluated by abundant simulation experiments.

The rest of this study is constructed as follows. Section II presents a concise review of related works. Section III presents system architecture, designs network and SC models. One multi-objective optimization problem model for VNF placement with multiple instances and traffic routing is derived and presented in Section IV. Section V proposes a distributed algorithm to solve above problem. Simulation results and analyses are demonstrated in Section VI, and many conclusions are obtained in Section VII.

II. RELATED WORK

There have been a significant amount of works done recently regarding VNF placement in IoT network. Authors in [9] design an algorithm to derive the optimal number of VNF instances of mobile network core elements. Taking QoE requirement into account, authors in [10] focus on VNF placement to support service differentiation among users, then present an Integer Linear Programming model (ILP) and a heuristic algorithm to minimize the associated SC deployment cost. Authors in [11] derive an analytical approach to characterize the latency performance and design a reasonable service traversal algorithm to obtain the optimal traffic routing path. In [12], authors formulate a mixed-integer Linear Programming model to efficiently place VNF instances and schedule service flows in distributed server clusters or data centers, and then derive a relaxation-based method to minimize both placement and communication cost with rational computational complexity. Additionally, authors in [13]–[16] also present some methods to guarantee the QoE and resource utilization in SC mapping. However, above methods ignore the effect of multiple VNF instances, which limits the effective improvement of cost efficiency. Moreover, VNF placement with single instance has less help for network load balancing.

Several studies start to solve the optimal number of instances for one kind of VNF in the process of SC mapping. Authors in [17] firstly propose to address the SC mapping with the optimal number of instances to minimize the deployment and communication resource consumption. They also propose an ILP model, a column-generation (CG) model as well as a two-phase (TP) model to solve this problem. Authors in [18] research the effect of VNF replicas on maintaining load balanced, and further present three optimization methods involving genetic algorithm, Linear Programming model and random placement algorithm to solve the optimal number, placement of VNF instances and traffic routing. Authors in [19] consider multiple instances in SC mapping and propose an algorithm to optimize the problem of scaling, placement of virtual network services and traffic routing. They present a mixed integer programming model, a customized heuristic algorithm to solve the problem. Additionally, authors in [20]–[23] also design many kinds of methods in terms of VNF placement with multiple instances. Although considering multiple VNF instances, those algorithms tend to solve NP-hard problem with centralized methods. The centralized methods have strict demands of computing resource for one machine, which reduces computational efficiency in engineering practice.

In summary, we propose an approach to optimize the placement of VNF with multiple instances and traffic routing. We build a multi-objective optimization problem model aiming to maintain load balanced and guarantee QoE requirements. As for the defect of centralized algorithms, we design VPMIA to solve above combinatorial optimization problem.

III. SYSTEM MODEL AND PROBLEM FORMULATION

A. SYSTEM ARCHITECTURE

In this section, we design a IoT service provisioning architecture based on NFV and Software Defined Network (SDN) in Fig. 1. The architecture mainly includes

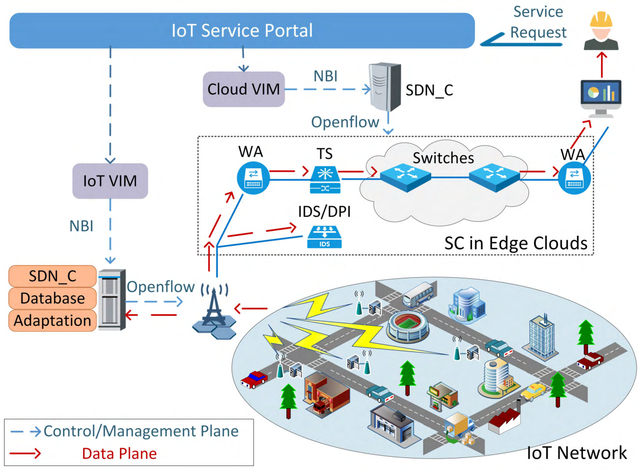

FIGURE 1. IoT service provisioning architecture.

following components. The IoT Service Portal is responsible to receive the requests and supply various services for IoT users. The Cloud Virtualized Infrastructure Manager (VIM) orchestrates diverse network function modules according to the rule generated by the Business Support System or Operation Support System. This component is also in charge of QoS, register, security and authentication. The Cloud SDN controller (SDN_C) consists of two parts: core SDN_C and flow SDN_C. The former is responsible for the VNF placement and management (i.e., service deployment and migration), and the latter takes charge of traffic scheduling among edge cloud servers. The IoT VIM is able to manage components and resources in the IoT network. The IoT SDN_C executes the software-defined control plane of the IoT work. The IoT network includes infrastructure resources of Radio Access Network (RAN), IoT-based sensors and actuators. It takes charge of data collection. Then the data is processed by ordered VNFs (i.e., WA, IDS, DPI, TS) deployed in edge clouds and dispatched to end users at last.

B. NETWORK MODEL

The substrate network of edge clouds is represented by weighted undirected graph $G = (N, L)$, where N and L denote nodes and links respectively. This paper classifies nodes into two types: 1) switch nodes forwarding traffic; 2) server nodes hosting virtual machines and represented by $v \in V$. Server v used to place VNF instances in mobile edge clouds has CPU computing and memory resources, which are represented by $Cap_{cpu}(v)$ and $Cap_{mem}(v)$ respectively. Physical link l_{ij} connecting nodes i and j has maximum data transmission rate b_{ij} and transmission delay d_{ij}. The specific definition of variables in this paper is shown in Table 1.

C. SC MODEL

SC request s in IoT network is represented by a triple $s = \{(v_{so}, v_{de})_s, F_s, r_s\}$, where $(v_{so}, v_{de})_s$ represents the source and destination node pair of s. F_s represents the specific SC information including the types, order and interconnection of VNFs. This paper assumes that the SC comprises non-replicable VNFs (i.e., PGW, SGW) and a series of ordered VNFs (i.e., WA, IDS, DPI, TS) which can be placed with multiple instances in mobile edge clouds. The non-replicable VNFs can only be assigned to dedicated clouds to generate a SC request s with a data transmission rate r_s. The required CPU, memory resources and processing delay of VNF $f \in F_s$ are represented by cpu_f, mem_f and d_f. The virtual link between VNFs u and w is represented by l^v_{uw}.

TABLE 1. Definition of variables.

Symbol	Description
$G = (N, L)$	weighted undirected graph
$v \in V$	cloud server nodes
$f \in F_s$	VNF f in SC s
r_s	data transmission rate of SC request s
$Cap_{cpu}(v)$	CPU resource in server v
$Cap_{mem}(v)$	memory resource in server v
$Cap(v)_{remain}$	weighted sum of remaining resources in server v
cpu_f	required CPU by VNF f
mem_f	required memory resource by VNF f
d_f	processing delay of VNF f
l_{ij}	physical link
l^v_{uw}	virtual link between VNFs u and w
b_{ij}	data transmission rate of l_{ij}
b^{ij}_{remain}	remaining transmission rate of l_{ij}
d_{ij}	transmission delay of l_{ij}
Φ_v	load balancing factor of server v
Θ_{ij}	load balancing factor of link l_{ij}
$\alpha_1, \beta_1, \lambda_1$	adjustment parameters for Φ_v
$\alpha_2, \beta_2, \lambda_2$	adjustment parameters for Θ_{ij}
D_s	minimum delay requirement
c_1, c_2, c_3	unit price of CPU, memory and transmission rate
$Cost_{total}$	total network cost
$(v_{so}, v_{de})_s$	source and destination node pair of s
z_p, z_m, z_d, z_l	weights of CPU, memory, delay and load balancing factor
Binary Variable	**Description**
$x^s_{v,f}$=1	VNF f in s is mapped to server v
$x^s_{v,f'}$=1	VNF replica f' of f in s is mapped to server v
$y^s_{ij,uw}$=1	l^v_{uw} in s is mapped to physical link l_{ij}

D. PROBLEM FORMULATION

The objective of this paper is to determine the number, placement of VNF instances and traffic routing for serving the IoT users while improving cost efficiency and guaranteeing load balancing. We assume that the order and interconnection of VNFs in the SC are fixed, but the number of instances can be adjusted by methods such as VNF replication. As shown in Fig. 2, the data transmission rate b_{ij} and transmission delay d_{ij} of links are set as 30Mbps and 10ms. Service traffic is divided evenly on multiple paths. Moreover, we let variance of link resource usage rate to denote load balancing level. For the SC request s composing of VNF1, VNF2, VNF3 with r_s = 15Mbps, scheme (a) completes the mapping by placing one VNF2 and one VNF3 on clouds. As a comparison, scheme (b) places VNF2 and VNF3 with 3 instances in mobile edge clouds. The forwarding cost, placement cost, delay and variance of link resource usage rate of two schemes are shown in Table 2.

In summary, scheme (a) only expends the cost of three instances, but it increases forwarding cost and transmission delay. Additionally, scheme (a) makes the load too

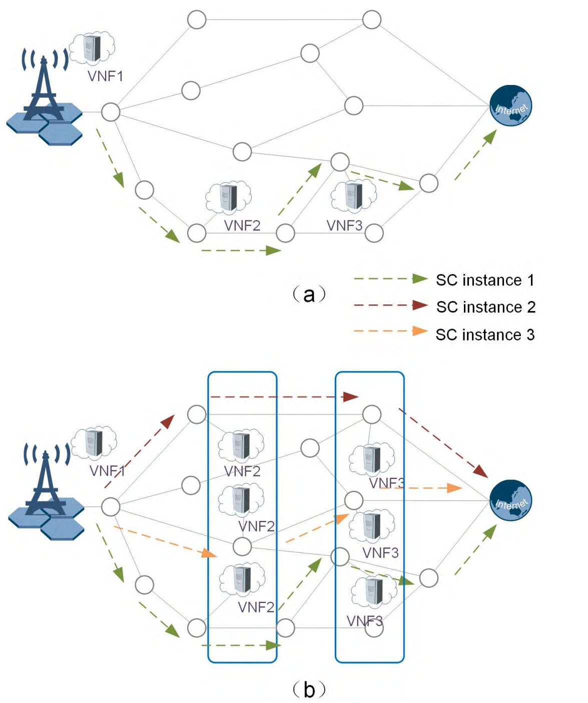

FIGURE 2. VNF placement with multiple instances.

TABLE 2. Performance of two schemes.

scheme	forwarding cost	placement cost	delay	variance
(a)	120Mbps	3	60ms	0.1
(b)	60Mbps	7	40ms	0.006

concentrated and easily leads to network congestion. As a contrast, scheme (b) decreases traffic forwarding cost and balances network load at the expense of additional placement cost. Thus, the number of instances have important impact on cost efficiency, QoE guarantee and load balancing.

IV. VNF PLACEMENT WITH MULTIPLE INSTANCES

This paper defines three binary variables to describe VNF placement with multiple instances and traffic routing.

- $x^s_{v,f}$: $x^s_{v,f}$=1 indicates that VNF f in s is mapped to physical node v, otherwise $x^s_{v,f}$=0;
- Particularly, $x^s_{v,f'}$=1 indicates that the replica f' of VNF f in s is mapped to physical node v, otherwise $x^s_{v,f'}$=0;
- $y^s_{ij,uw}$: $y^s_{ij,uw}$=1 indicates that virtual link l^v_{uw} in s is mapped to physical link l_{ij}, otherwise $y^s_{ij,uw}$=0.

A. COST AND LOAD BALANCING FACTORS

VNF instances need CPU, memory and other physical resources in cloud servers. Load balancing is taken into account, since maintaining the load balanced among servers and links can avoid traffic congestion and further improve network cost efficiency. Therefore, two load balancing factors including Φ_v and Θ_{ij} are proposed to indicate load status of network, and their values have negative relations to the remaining resources. Φ_v is given by

$$\Phi_v = \frac{\alpha_1}{Cap(v)_{remain} + \beta_1} + \gamma_1 \tag{1}$$

where $\alpha_1, \beta_1, \lambda_1$ are adjustment parameters and used to adjust the value of Φ_v in the process of cost calculation. $Cap(v)_{remain}$ represents weighted sum of remaining CPU and memory in v. It is given by

$$\begin{aligned} Cap(v)_{remain} = z_p \cdot (Cap_{cpu}(v) - \sum_{s\in S}\sum_{f\in F_s} x^s_{v,f} \cdot cpu_f) \\ + z_m \cdot (Cap_{mem}(v) - \sum_{s\in S}\sum_{f\in F_s} x^s_{v,f} \cdot mem_f) \end{aligned} \tag{2}$$

where z_p and z_m represent the weights of remaining CPU and memory resources, and $z_p + z_m = 1$. Therefore related cost of consuming server resources is given by

$$cost(server) = \sum_{s\in S}\sum_{v\in V}\sum_{f\in F_s} x^s_{v,f}(c_1 \cdot cpu_f + c_2 \cdot mem_f) \cdot \Phi_v \tag{3}$$

The unit price of CPU and memory resource are denoted by c_1 and c_2 respectively. From (3) we can get that $cost(server)$ consists of two parts: Φ_v and the fixed unit price. Next, we consider forwarding cost in traffic routing. The load balancing factor Θ_{ij} is given by

$$\begin{aligned} \Theta_{ij} &= \frac{\alpha_2}{b^{ij}_{remain} + \beta_2} + \gamma_2 \\ &= \frac{\alpha_2}{b_{ij} - \sum_{s\in S}\sum_{i,j\in N}\sum_{u,w\in F_s} r_s \cdot y^s_{ij,uw} + \beta_2} + \gamma_2 \end{aligned} \tag{4}$$

where $\alpha_2, \beta_2, \lambda_2$ are adjustment parameters and used to adjust the value of Θ_{ij} in the process of cost calculation. c_3 indicates unit price of transmission rate of links. Therefore the cost of traffic forwarding is given by

$$cost(link) = c_3 \sum_{s\in S}\sum_{i,j\in N}\sum_{u,w\in F_s} y^s_{ij,uw} \cdot r_s \cdot (z_l \cdot \Theta_{ij} + z_d \cdot d_{ij}) \tag{5}$$

$(z_l \cdot \Theta_{ij} + z_d \cdot d_{ij})$ represents the weighted sum of Θ_{ij} and delay d_{ij}, and $z_l + z_d = 1$. $cost(link)$ consists of three parts: Θ_{ij}, the fixed delay d_{ij} and unit price. It can be seen from (3) and (5) that links or nodes with larger remaining resource have relatively lower cost. Therefore, links or nodes of this type are more likely to be selected. Through Φ_v and Θ_{ij}, our SC mapping scheme is able to balance the network load, and then improve network resource utilization.

The total cost $Cost_{total}$ in the process of SC mapping is given by

$$Cost_{total} = cost(server) + cost(link) \tag{6}$$

B. CAPACITY CONSTRAINTS

$$C_1 : \sum_{s\in S}\sum_{f\in F_s} x^s_{v,f} \cdot cpu_f \le Cap_{cpu}(v), \quad \forall v$$

$$C_2 : \sum_{s\in S}\sum_{f\in F_s} x^s_{v,f} \cdot mem_f \le Cap_{mem}(v), \quad \forall v \tag{7}$$

$$C_3 : \sum_{s\in S}\sum_{u,w\in F_s} y^s_{ij,uw} \cdot r_s \le b_{ij}, \quad \forall i,j \tag{8}$$

$$C_4 : x^s_{v_0,f} + x^s_{v_0,f'} \le 1, \quad \forall f \in F_s \tag{9}$$

$$C_5 : 0 \le \sum_{v \in V} x^s_{v,f'} \le f_{max}, \quad \forall f \tag{10}$$

$$C_6 : \sum_{i,j \in N} y^s_{ij,uw} - \sum_{i,j \in N} y^s_{ji,uw} = \begin{cases} 1, & if\ i \in v_{so} \\ -1, & if\ i \in v_{de} \\ 0, & otherwise \end{cases} \tag{11}$$

The server v can continue to host VNF instances only if it has enough remaining CPU and memory resources. This constraint is shown as (7).

Similarly, the link l_{ij} can continue to forward service traffic only if it has enough remaining data transmission rate. This constraint is shown as (8).

We assume that the VNF instance f and its replica f' cannot be placed on the same server, because this placement increases the complexity of the mapping. On the other hand, two identical VNF instances placed on same server can be replaced by one. This constraint is shown as (9).

A VNF instance cannot be replicated indefinitely, so the maximum number of replicas for one VNF is constrained by (10).

As shown in (11), according to flow conservation constraint, the sum of directional traffic of other nodes is 0 except source node and destination node in the process of routing.

C. DELAY CONSTRAINT

This paper considers that SC request s has requirement of minimum delay D_s. The end-to-end delay mainly includes processing delay on VNFs and transmission delay on links. The SC mapping scheme is valid only if its end-to-end delay meets requirement.

$$C_7 : \sum_{v \in V} \sum_{f \in F_s} x^s_{v,f} \cdot d_f + \sum_{i,j \in N} \sum_{u,w \in F_s} y^s_{ij,uw} \cdot d_{ij} \le D_s \tag{12}$$

D. MULTI-OBJECTIVE OPTIMIZATION MODEL

Given the above, a multi-objective optimization problem model aiming to improve cost efficiency and guarantee load balancing is built as (13).

$$\begin{aligned} &\boldsymbol{Min}\ \{cost(server) + cost(link)\} \\ &s.t. \begin{cases} C_1, C_2, C_3, C_4, C_5, C_6, C_7 \\ C_8 : x^s_{v,f}, x^s_{v,f'}, y^s_{ij,uw} \in \{0, 1\} \quad \forall v, f, f', i, j, u, w \end{cases} \end{aligned} \tag{13}$$

V. ALGORITHM DESCRIPTION

SC mapping problem involves the number, placement and routing of VNFs, so it is an NP-hard combinatorial optimization problem. Recently, a large number of centralized algorithms have been utilized to solve this problem [24], [25]. However, centralized algorithms require huge computing and memory resources for a single physical machine, and have lower computational efficiency. In contrast, distributed algorithms can decentralize tasks to improve computational efficiency. Authors in [26] have proved that the combinatorial optimization problem can be transferred according to Log-Sum-Exp approximation technique into a problem which can be solved in a distributed way. Further, algorithms on the basis of the time-reversible Markov chain model can obtain near-optimal solution of this problem efficiently [27]. Therefore, we approximate the optimization problem model of formula in (13) and propose a VNF placement with multiple instances algorithm based on Markov chain.

A. LOG-SUM-EXP APPROXIMATION METHOD

Definition of available action set $A = \{X, Y\}$:

$$X = \{\{x^s_{f,v}\}, x^s_{f,v} \in \{0, 1\}, s \in S\} \tag{14}$$

X is defined as the set of available actions for VNF placement (including VNF replicas) in SC mapping. If X is determined, the routing between VNFs can be determined by the shortest path algorithm.

$$Y = \left\{ \left\{ y^s_{ij,uw} | x \right\}, y^s_{ij,uw} \in \{0, 1\}, s \in S \right\} \tag{15}$$

Y is defined as the set of available actions for traffic routing between VNFs. Therefore, the available action set for VNF placement and traffic routing is defined as $A = \{X, Y\}$.

Definition of objective function $Cost$: Let $a \in A$ denote a specific action, and $Cost(a)$ denote the network total cost according to (6) after taking action a. Therefore the objective function is defined as follow:

$$\min_{a \in A} Cost(a) \tag{16}$$

Its optimal value is the same as that of following problem.

$$\begin{aligned} &\min_{p>0} \sum_{a \in A} p(a) \cdot Cost(a) \\ &s.t. \sum_{a \in A} p_a = 1 \end{aligned} \tag{17}$$

where p_a represents the percentage of execution time for action a. Next, we introduce Markov approximation technique to approximate the formula in (16). First, a differentiable function is defined.

$$\delta_\beta(\boldsymbol{Cost}) = \frac{1}{\beta} \cdot \log(\sum_{a \in A} \exp(Cost(a))) \tag{18}$$

where β is an integer constant, and $\boldsymbol{Cost} \triangleq [Cost(a), a \in A]$. The objective function is approximated by the formula in (18) and expressed as follow.

$$\min_{a \in A} Cost(a) \approx \frac{1}{\beta} \cdot \log(\sum_{a \in A} \exp(\beta \cdot Cost(a))) \tag{19}$$

Above equation is called convex and closed Log-Sum-Exp approximation for objective function. What's more, the approximation difference is less than $\frac{1}{\beta} \log |A|$. The proof is as follows.

Proposition 1. For a positive constant β and n non-negative real variables $h_1, h_2, \ldots, h_n$, there are:

$$\max_{i=1,2,\ldots,n} h_i \leq \frac{1}{\beta} \cdot \log(\sum_{i=1} \exp(\beta \cdot h_i)) \leq \max_{i=1,2,\ldots,n} h_i + \frac{1}{\beta} \cdot \log n \tag{20}$$

Proof: Let $h_{\max}$ represent the maximum value in $h_1, h_2, \ldots, h_n$, and then bring $\sum_{i=1} \exp(\beta \cdot h_i) \geq \exp(\beta \cdot h_{\max})$ and $\sum_{i=1} \exp(\beta \cdot h_i) \leq n \cdot \exp(\beta \cdot h_{\max})$ into (20). Therefore, the *Proposition 1* can be proved.

Since the problem in (16) is the minimum function, so we define H as a large enough real number, and bring $h'_i = H - h_i$ into (20).

$$\min_{i=1,2,\ldots,n} h'_i \geq \frac{1}{\beta} \cdot \log(\sum_{i=1} \exp(\beta \cdot h'_i)) \geq \min_{i=1,2,\ldots,n} h'_i - \frac{1}{\beta} \cdot \log n \tag{21}$$

Hence, the maximum approximation difference is less than $\frac{1}{\beta} \log |A|$, and

$$\min_{i=1,2,\ldots,n} h_i{}' = \lim_{\beta \to \infty} \frac{1}{\beta} \cdot \log(\sum_{f \in F} \exp(\sum_{s \in S} \beta \cdot h_i{}')) \tag{22}$$

For the approximation function $\delta_\beta(\boldsymbol{Cost})$, we calculate its conjugate function $\delta^*{}_\beta(\boldsymbol{p})$, and it is given by

$$\delta^*_\beta(\boldsymbol{p}) = \begin{cases} \frac{1}{\beta} \cdot \sum_{a \in A} p_a \cdot \log(p_a), & \text{if } \boldsymbol{p} \geq 0, \mathbf{1}^T \boldsymbol{p} = 1 \\ \infty, & \text{otherwise} \end{cases} \tag{23}$$

Since $\delta_\beta(\boldsymbol{Cost})$ is closed and convex function, the conjugate of $\delta^*{}_\beta(\boldsymbol{p})$ is still $(\delta_\beta(\boldsymbol{Cost}))$. We continue to calculate the conjugate of $\delta^*{}_\beta(\boldsymbol{p})$ and then get the function in (24). Therefore, $\delta_\beta(\boldsymbol{Cost})$ can be solved by solving problem in (24) since they have same optimal solution.

$$\begin{aligned} &\min_{\boldsymbol{p}>0} \sum_{a \in A} p_a \cdot Cost(a) + \frac{1}{\beta} \cdot \sum_{a \in A} p_a \cdot \log(p_a) \\ &s.t. \sum_{a \in A} p_a = 1 \end{aligned} \tag{24}$$

Proposition 2. Problem in (24) belongs to the convex optimization problem.

Proof: Above proposition is proved by the second-order necessary and sufficient conditions of the convex function. Hessian matrix of the objective function is positive in the value space, and all constrains of problem in (24) are linear. Therefore it is a convex optimization problem.

Based on *Proposition 2*, the optimal solution p^*_a for the problem in (24) meets the Karush-Kuhn-Tucher (KKT) conditions, and it can be obtained by the Lagrange multiplier method as follows.

$$\begin{aligned} &\sum_{s \in S} Cost(a) - \frac{1}{\beta} \log p^*(\boldsymbol{Cost}) + \frac{1}{\beta} + \lambda = 0, \quad \forall a \in A \\ &\sum_{a \in A} p^*(\boldsymbol{Cost}) = 1 \quad \lambda \geq 0 \end{aligned} \tag{25}$$

where λ is a Lagrange multiplier. Then:

$$p^*_a(\boldsymbol{Cost}) = \frac{\exp(-\beta \cdot Cost(a))}{\sum_{a \in A} \exp(-\beta \cdot Cost(a'))}, \quad \forall a \in A \tag{26}$$

According to $p^*_a(\boldsymbol{Cost})$ in (26), the optimal time ratio of the action a can be determined, and then the problem in (16) can be solved.

B. TIME-REVERSIBLE MARKOV CHAIN MODEL

In this section, we try to construct the Markov chain model based on state space A and stationary distribution space p^*_a as shown in (26). This idea is mainly supported by following two theories:

1) For any probability distribution of product-form $p^*_a(\boldsymbol{Cost})$, there is at least one continuous-time time-reversible Markov chain with stationary distribution $p^*_a(\boldsymbol{Cost})$ [28].

2) For each time interval, the distribution of forward motion is the same as that of backward motion, so time-reversible Markov chain have better ergodicity for state space and can converge to the ideal distribution within an acceptable period of time.

Therefore, we design a time-reversible Markov chain. $a \in A$ involves VNF placement and traffic routing scheme, and represents the state in Markov chain. $q_{a,a'}$ represents the non-negative transition rate from state a to a'. This transition means that the current scheme of VNF placement with multiple instances and traffic routing is transferred to a new scheme. We assume that any two states can be arbitrarily reachable and satisfy the following balance equation.

$$p^*{}_a \cdot q_{a,a'} = p^*{}_{a'} \cdot q_{a',a}, \quad \forall a, \; a' \in A \tag{27}$$

Let $q_{a,a'}$ be positively correlated to the cost difference between state a and a'. It is expressed as follows.

$$q_{a,a'} = \exp(\frac{1}{2} \cdot \beta \cdot (Cost(a) - Cost(a')) - \tau) \tag{28}$$

where τ is a non-negative constant and used to avoid $q_{a,a'}$ being too large. It can be seen that the system tends to transfer to a state with less network cost.

We let each SC run on a separate continuous clock and waits for cost-dependent timer before it changes its local action, so the transition between a and a' can be done by changing SC's local action. By this means, this time-reversible Markov chain will be executed in the distributed way. With the convergence of Markov chain, the optimal actions will be taken with the maximal percentage of time based on $p^*_a(\boldsymbol{cost})$. Therefore, system will get the near-optimal solution.

C. VNF PLACEMENT WITH MULTIPLE INSTANCES ALGORITHM BASED ON MARKOV CHAIN

This section designs VPMIA based on Markov chain. The algorithm mainly includes procedure initialization, procedure and timer set, procedure transition, procedure reset. The transition process of each SC observes the general state machine in Fig. 3.

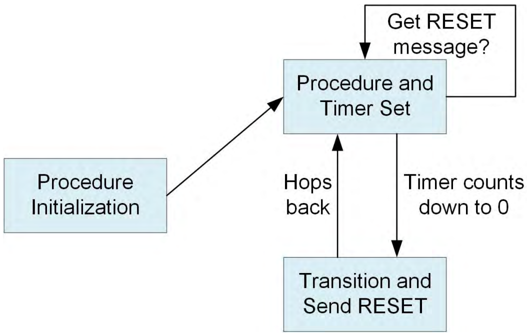

FIGURE 3. General state machine.

Algorithm 1 Procedure Initialization

Input: $G = (N, L)$; b_{ij}; d_{ij}; $Cap_{cpu}(v)$; $Cap_{mem}(v)$; $s = \{(v_{so}, v_{de})_s, F_s, r_s\}$;
Output: Initialization scheme;
Weights of links are initialized as $c_3 \cdot (z_l \cdot \Theta_{ij} + z_d \cdot d_{ij})$
for $s \in S$ **do**;
  Calculate the feasible path between $(v_{so}, v_{de})_s$ by shortest path algorithm;
  Calculate available servers in current path;
  if the number of available servers is not enough **then**
    Search for available servers closest to current path;
    Extend path to new servers;
  end if
  Place randomly VNF instances $f \in s$ in the available servers;
  Get the action set $\{x^s_{f,v}\}$ and $\{y^s_{ij,uw}|x\}$ of s;
  Calculate $Cost(a)$ based on (6);
end for
return $\{x^s_{f,v}\}$, $\{y^s_{ij,uw}|x\}$, $Cost(a)$;

1) PROCEDURE INITIALIZATION

The algorithm first allocates dedicated threads for each SC in the current system. For an incoming SC s, its initialization of VNF placement and traffic routing is shown as follows.

1) The weights of links are set to be the weighted sum of delay and load balancing factor, and then the shortest path according to source node and destination node pair $(v_{so}, v_{de})_s$ is obtained.

2) If there are enough servers, VNF instances needed by s are randomly placed in the available servers on the current path according to constraints.

3) If the number of servers is insufficient, the current path will be extended to the nearest servers until the placement of all VNF instances is completed.

4) At last, we calculate the cost $Cost_s(a)$ under the current state a. The procedure is shown in Alg.1.

2) PROCEDURE AND TIMER SET

For current SC s with state a, algorithm increases the number of VNF instances by replication. The steps are as fellows.

Algorithm 2 Procedure and Timer Set

Input: $G = (N, L)$; b_{ij}; d_{ij}; $Cap_{cpu}(v)$; $Cap_{mem}(v)$; $s = \{(v_{so}, v_{de})_s, F_s, r_s\}$;
Output: $\sigma_{f,s}$; $\{x^s_{v,f'}\}$; $\{y|x'\}$
for $s \in S$ **do**;
  Choose a VNF f randomly in s;
  Calculate $\sigma_{f,s}$ for s;
  if $|\sigma_{f,s}| \geq 1$ **then**
    Choose a replication scheme $x^s_{v,f'} = 1$ form $\sigma_{f,s}$ randomly;
    Compute the shortest path $(y|x')$ form f' to its adjacent instances $f+1$ and $f-1$;
    Calculate the new $Cost(a')$ based on $x^s_{v,f'}$ and $(y|x')$;
    Calculate the timer T_s according to (29);
  end if
end for
return $\sigma_{f,s}$; $\{x^s_{v,f'}\}$; $\{y|x'\}$;

1) The algorithm selects randomly a VNF instance f to replicate. Let $\sigma_{f,s} = \{x^s_{v_1,f'} = 1, x^s_{v_2,f'} = 1, .., \}$ represents the set of optional replication actions for f.

2) If $\sigma_{f,s}$ is empty, it means that current VNF instance f cannot be replicated, or there are no available servers to place replicas. Then the algorithm ends.

3) If not, algorithm selects a replication scheme $x^s_{v,f'} = 1$ form $\sigma_{f,s}$ randomly and takes corresponding actions. Now, f has a new replica f'.

4) Algorithm computes the shortest path $(y|x')$ form f' to its adjacent instances $f+1$ and $f-1$. Then, the cost $Cost_s(a')$ of current state a' of s can be obtained based on (6).

5) Algorithm generates a random exponentially distributed timer T_s to determine whether to update state a' from a. Meanwhile, T_s begins to count down. T_s is defined as follow.

$$T_s = \frac{1}{|F_s| \cdot |\sigma_{f,s}|} \cdot \exp(\tau - \frac{1}{2} \cdot \beta \cdot (Cost(a) - Cost(a'))) \tag{29}$$

3) PROCEDURE TRANSITION

Algorithm chooses the $x^s_{v,f'}$ and corresponding paths as its replication scheme of f if T_s expires. Then **Procedure and Timer Set** will be executed repeatedly, and RESET will be broadcast to notify state update.

4) PROCEDURE RESET

If SC s gets a RESET message, T_s will be updated on the basis of new $Cost(a)$. Therefore the main Procedure of the VPMIA is shown in Alg.3.

D. CONVERGENCE ANALYSIS

This section analyzes the convergence time T_{con} of above time-reversible Markov chain. According to the theorem

Algorithm 3 VPMIA

Input: $G = (N, L)$; b_{ij}; d_{ij}; $Cap_{cpu}(v)$; $Cap_{mem}(v)$; $s = \{(v_{so}, v_{de})_s, F_s, r_s\}$;
Output: $\{x^s_{v,f}\}$; $\{y^s_{ij,uw}\}$; $\{y|x'\}$;
Execute **Alg.1**;
Execute **Alg.2**;
while system is running **do**
  /*Procedure Transition*/
  for $s \in S$ **do**;
    Choose a VNF f randomly in s;
    Obtain its current placement scheme: $\{x^s_{v,f}\}$ and $\{y^s_{ij,uw}\}$;
    Choose a new scheme $x^s_{v,f'}$ in $\sigma_{f,s}$;
    Calculate $Cost(a')$ and T_s;
    if T_s expires **then**
      Choose $x^s_{v,f'}$ as a replication scheme of f;
      Compute the shortest path $(y|x')$ form f' to its adjacent instances $f+1$ and $f-1$;
      Execute **Alg.2;**
      Spread RESET message to other SC to notify state update;
    end if
    /*Procedure RESET*/
    if SC s receives the RESET message **then**
      Put $x^s_{v,f'}$ as replication scheme f;
      Replace $Cost(a)$ with $Cost(a')$;
      Clear and generate a new timer T_s;
    end if
  end for
end while
return $\{x^s_{v,f}\}$; $\{y^s_{ij,uw}\}$; $\{y|x'\}$;

in [26], for $\beta \in (0, +\infty)$, T_{con} is given as follows.

$$T_{con}(o) \geq \frac{\exp(\tau - \frac{1}{2} \cdot \beta \cdot (cost_{\max} - cost_{\min}))}{2 \cdot \eta} \cdot \ln \frac{1}{2 \cdot o} \tag{30}$$

and

$$T_{con}(o) \leq 2 \cdot \eta \cdot \zeta^2 \cdot \exp(\frac{3}{2} \cdot \beta \cdot (cost(a) - cost(a')) + \tau) \cdot (\ln \frac{1}{2 \cdot o} + \frac{1}{2} \cdot \ln \xi + \frac{1}{2} \cdot \beta \cdot (cost(a) - cost(a'))) \tag{31}$$

where o represents the optimality difference, and $\sigma_{f,s}$ represents the schemes of placement and traffic routing of VNF replica f'. Definition of ζ is shown as follow.

$$\xi = \prod\nolimits_{s \in S} |F_s| \cdot |\sigma_{f,s}| \tag{32}$$

It represents the number of all states in Markov chain, and $\eta = \sum_{s \in S} |F_s| \cdot |\sigma_{f,s}|$.

VI. EVALUATION ANALYSIS

A. SIMULATION SETTING

To reflect the effectiveness of proposed algorithm, we develop a network simulation platform based on Java language. The substrate network consisting of 30 nodes (10 servers and 20 forwarding nodes) and 50 links is set up to represent mobile edge cloud network by Java language. The maximum data transmission rate of links is fixed to 1Gbps. The CPU and memory resource of servers are set as 32 and 100-200GB respectively. Each SC needs 2-4 VNFs and its data transmission rate is 20-50 Mbps. Each VNF requires 2-4 CPU and 5-10GB memory resource. The specific parameters are given in Table 3.

TABLE 3. Specific experimental parameters.

The number of servers	10
The number of forwarding nodes	20
The number of links	50
The CPU of one server	32
The memory resource of server	100-200GB
The transmission rate of link	1Gbps
The required VNFs of SC	2-4
The CPU required by one VNF	2-4
The memory required by one VNF	5-10GB
The data transmission rate of one SC	20-50Mbps
$\alpha_1, \beta_1, \lambda_1$	0.1, 0.1, 0.1
$\alpha_2, \beta_2, \lambda_2$	0.1, 0.1, 0.1
z_p, z_m, z_d, z_l	0.5

We take random placement algorithm (RPA) and single-path algorithm (SPA) as the comparisons. Random placement algorithm first gets the feasible paths between source and destination nodes under the constraints. Then it selects a path randomly and places VNF instances on this path. To balance network load, RPA will repeat above selection and placement until certain VNF meets the maximum number of replicas. SPA is the same as the Procedure Initialization in VPMIA, but it does not replicate any VNF instances.

B. RESULT ANALYSIS

Several indicators are selected to evaluate performance of three algorithms.

1) NETWORK COST

Fig. 4 shows total cost of three algorithms with the different number of SCs. Costs of SPA and RPA are slightly higher than that of VPMIA when the number of SCs is less than 40, but they are not much different. Note that the cost of SPA is slightly lower than RPA at 20-40, because it supplements only one shortest path, which has certain advantages when dealing with a small amount of SCs. However, as the number of SCs increases, total costs of SPA and RPA are significantly higher than that of VPMIA. Taking the SCs = 120 as an example, total costs of the two are 31% and 22% higher than that of VPMIA respectively. Because VPMIA reduces forwarding and placement costs by planning the number of VNF instances, and solves the multi-objective optimization problem by a distributed manner, which greatly improves its resource utilization. Conversely, SPA ignores the effects of multiple instances. Although RPA increases the number of

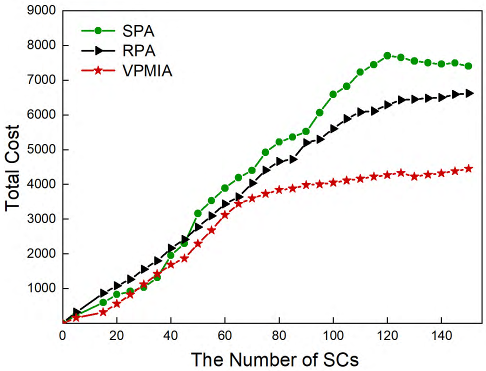

FIGURE 4. Total cost.

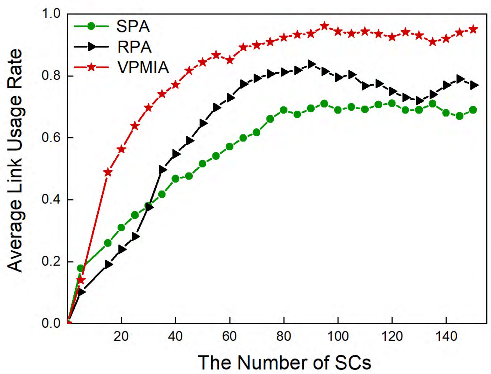

FIGURE 5. Average link usage rate.

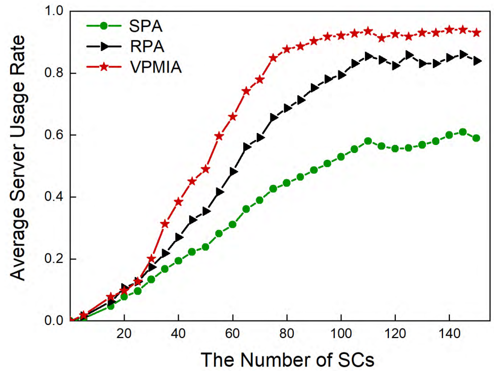

FIGURE 6. Average server usage rate.

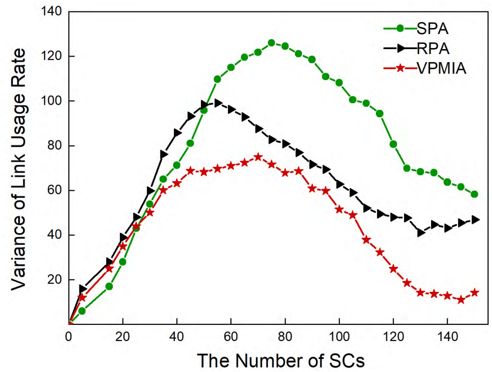

FIGURE 7. Variance of link usage rate.

VNF instances, its random selection and placement cannot guarantee the resource utilization.

2) AVERAGE LINK USAGE RATE

Fig. 5 shows average link usage rates of three algorithms. The average link usage rates of the three algorithms are basically equal when the number of SCs is less than 20, because the number of SCs at this time is small and three algorithms can complete mapping well. As the number of SCs increases, the advantages of VPMIA become larger and its average usage rate is significantly higher than the other two. Taking SCs = 80 as an example, the average usage rate of VPMIA is 51% and 27% higher than those of SPA and RPA respectively. This is because VPMIA designs the link load balancing factor to avoid link congestion. On the other hand, the customized placement with multiple instances reduces the forwarding distance of traffic, so the link resource can be utilized effectively.

3) AVERAGE SERVER USAGE RATE

Average server usage rates of three algorithms are shown in Fig. 6. The abscissa is the number of SCs, and the ordinate is average server usage rate. As can be seen from above figure, average server usage rate of VPMIA is always higher than the other two algorithms. Although this advantage is not obvious when the number of SCs is small, it is significantly higher than the other two algorithms when dealing with a large amount of SCs, since VPMIA guarantees load balancing of SCs by planning the number of instances reasonably. Meanwhile, the customized placement of VNF instances makes server resource be utilized maximally. Taking SCs = 100 as an example, average server usage rate of VPMIA is 47% and 16% higher than those of SPA and RPA respectively.

4) VARIANCE OF LINK USAGE RATE

Variances of link usage rate can directly reflect the load balancing of all links in network. The comparison of three algorithms is shown in Fig. 7. When the number of SCs is less than 40, variances of the three algorithms are not much different, because network has enough resources to complete the mapping of SCs. Variances of RPA and VPMIA are slightly lower than that of RPA, because they consider multiple instances in the process of VNF placement. As the number of SCs increases, the usage rate variances of SPA and RPA are significantly higher than that of VPMIA. Taking SCs = 100 as an example, the variances of SPA and RPA are 61% and 73% higher than VPMIA respectively. VPMIA designs the link load balancing factor, and ensures the load balancing by selecting the link with more remaining resource as much as possible, so its variance is always smaller than SPA and RPA.

5) VARIANCE OF SERVER USAGE RATE

Fig. 8 shows the variances of server usage rate for three algorithms. The abscissa is the number of SCs and the ordinate is variance value of server usage rate. The usage rate variances

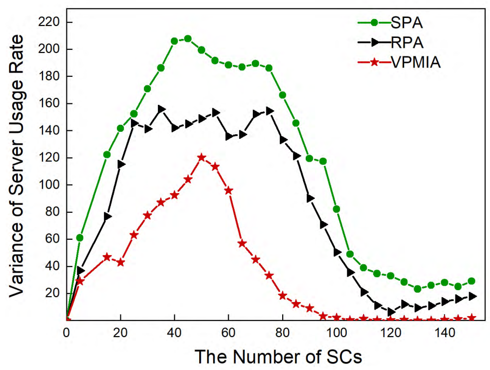

FIGURE 8. Variance of server usage rate.

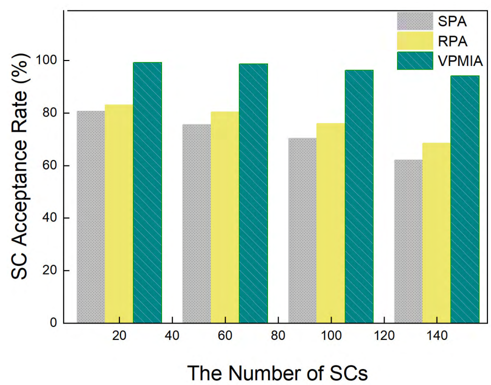

FIGURE 9. Acceptance rate.

of RPA and VPMIA are slightly lower than that of SPA when the number of SCs is less than 40. As the number of SCs continues to increase, usage rate variances of VPMIA is significantly higher than those of SPA and RPA. Taking SCs $= 80$ as an example, usage rate variances of the two are 81% and 72% higher than that of VPMIA respectively. VPMIA designs the server load balancing factor and places VNF with multiple instances. It ensures the load balancing of the server by selecting the SC with more remaining resources, so its variance is always smaller than SPA and VPMIA. Therefore, VPMIA can better guarantee the load balancing of servers compared to the other two algorithms.

6) ACCEPTANCE RATE

The SC acceptance rates of three algorithms are shown in Fig. 9. The abscissa is the number of SCs, the ordinate indicates the acceptance rate. VPMIA has higher acceptance rates than those of SPA and RPA when dealing with the different number of SCs. Because VPMIA puts optimal delay into objective function, and then optimize this index by customized placement of VNF instances. Meanwhile, the cost efficiency of VPMIA is apparently higher than that of SPA and RPA. Therefore, the SC acceptance rate of VPMIA is higher than that of SPA and RPA.

7) CONVERGENCE TIME

Fig. 10 shows the comparison of convergence time between VPMIA and ILP mode. ILP can obtain optimal solution by

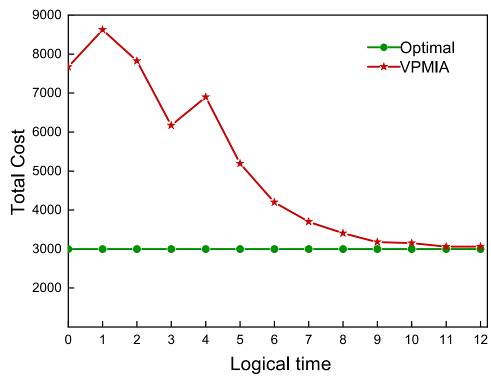

FIGURE 10. Convergence time.

using CPLEX optimizer in advance [29]. The abscissa is the logical time of algorithm execution, and the ordinate is total cost of two algorithms. At the beginning, solution derived by VPMIA has a low cost efficiency since it randomly selects schemes. As the convergence of VPMIA, its solution becomes closer to optimal solution derived by ILP model. When logical time reaches 9, VPMIA derives the near-optimal solution.

VII. CONCLUSION

In recent years, NFV has played a significant role in simplifying service deployment by placing and routing an ordered sequence of VNFs. On the other hand, mobile edge clouds globally distributed provide computing, memory resources to facilitate the customized placement of VNF instances at the mobile network edge, which has an important impact on reducing Capital Expenditure and Operation Expenditure. Therefore, we pay attention to the VNF placement with multiple instances to improve cost efficiency and balance network load. Then a multi-objective optimization problem model is built and converted to the problem which can be solved by distributed methods based on Markov approximation technique. As for above problem, a distributed algorithm is designed to solve the number, placement and routing of VNF instances. The simulation results demonstrate that the proposed algorithm has great advantages in terms of cost and computational efficiency, and can also better guarantee QoE and load balancing of network.

REFERENCES

[1] A. Basta, A. Blenk, K. Hoffmann, H. J. Morper, M. Hoffmann, and W. Kellerer, "Towards a cost optimal design for a 5G mobile core network based on SDN and NFV," *IEEE Trans. Netw. Service Manag.*, vol. 14, no. 4, pp. 1061–1075, Dec. 2017.

[2] L. Ma, X. Wen, L. Wang, Z. Lu, and R. Knopp, "An SDN/NFV based framework for management and deployment of service based 5G core network," *China Commun.*, vol. 15, no. 10, pp. 86–98, Oct. 2018.

[3] F. Z. Yousaf, M. Bredel, S. Schaller, and F. Schneider, "NFV and SDN—Key technology enablers for 5G networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2468–2478, Nov. 2017.

[4] S. Van Rossem *et al.*, "Deploying elastic routing capability in an SDN/NFV-enabled environment," in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, San Francisco, CA, USA, Nov. 2015, pp. 22–24.

[5] S. Van Rossem *et al.*, "A network service development kit supporting the end-to-end lifecycle of NFV-based telecom services," in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, Berlin, Germany, Nov. 2017, pp. 1–2.

[6] M. Bagaa, T. Taleb, A. Laghrissi, A. Ksentini, and H. Flinck, "Coalitional game for the creation of efficient virtual core network slices in 5G mobile systems," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 469–484, Mar. 2018.

[7] I. Benkacem, T. Taleb, M. Bagaa, and H. Flinck, "Optimal VNFs placement in CDN slicing over multi-cloud environment," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 616–627, Mar. 2018.

[8] A. Baumgartner, V. S. Reddy, and T. Bauschert, "Mobile core network virtualization: A model for combined virtual core network function placement and topology optimization," in *Proc. 1st IEEE Conf. Netw. Softwarization (NetSoft)*, London, U.K., Apr. 2015, pp. 1–9.

[9] A. Laghrissi, T. Taleb, and M. Bagaa, "Conformal mapping for optimal network slice planning based on canonical domains," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 519–528, Mar. 2018.

[10] P. Vizarreta, M. Condoluci, C. M. Machuca, T. Mahmoodi, and W. Kellerer, "QoS-driven function placement reducing expenditures in NFV deployments," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Paris, France, May 2017, pp. 1–7.

[11] Y.-J. Chen, L.-C. Wang, F.-Y. Lin, and B.-S. P. Lin, "Deterministic quality of service guarantee for dynamic service chaining in software defined networking," *IEEE Trans. Netw. Service Manag.*, vol. 14, no. 4, pp. 991–1002, Dec. 2017.

[12] L. Gu, X. Chen, H. Jin, and F. Lu, "VNF deployment and flow scheduling in geo-distributed data centers," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kansas City, MO, USA, May 2018, pp. 1–6.

[13] N. Huin, B. Jaumard, and F. Giroire, "Optimal network service chain provisioning," *IEEE/ACM Trans. Netw.*, vol. 26, no. 3, pp. 1320–1333, Jun. 2018.

[14] A. Leivadeas, M. Falkner, I. Lambadaris, and G. Kesidis, "Resource management and orchestration for a dynamic service chain steering model," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Washington, DC, USA, Dec. 2016, pp. 1–6.

[15] J.-J. Kuo, S.-H. Shen, H.-Y. Kang, D.-N. Yang, M.-J. Tsai, and W.-T. Chen, "Service chain embedding with maximum flow in software defined network and application to the next-generation cellular network architecture," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Atlanta, GA, USA, May 2017, pp. 1–9.

[16] M. Jalalitabar, E. Guler, D. Zheng, G. Luo, L. Tian, and X. Cao, "Embedding dependence-aware service function chains," *J. Opt. Commun. Netw.*, vol. 10, no. 8, pp. C64–C74, Aug. 2018.

[17] A. Gupta, B. Jaumard, M. Tornatore, and B. Mukherjee, "A scalable approach for service Chain mapping with multiple SC instances in a wide-area network," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 529–541, Mar. 2018.

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

[19] S. Draxler, H. Karl, and Z. Mann, "JASPER: Joint optimization of scaling, placement, and routing of virtual network services," *IEEE Trans. Netw. Service Manag.*, vol. 15, no. 3, pp. 946–960, Sep. 2018.

[20] F. Carpio, W. Bziuk, and A. Jukan, "Replication of virtual network functions: Optimizing link utilization and resource costs," in *Proc. 40th Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO)*, Opatija, Croatia, May 2017, pp. 521–526.

[21] F. Carpio, A. Jukan, and R. Pries, "Balancing the migration of virtual network functions with replications in data centers," in *Proc. IEEE/IFIP Netw. Oper. Manage. Symp. (NOMS)*, Taipei, Taiwan, 2018, pp. 1–8.

[22] J. Kong *et al.*, "Guaranteed-availability network function virtualization with network protection and VNF replication," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Singapore, Dec. 2017, pp. 1–6.

[23] S. G. Kulkarni, M. Arumaithurai, K. K. Ramakrishnan, and X. Fu, "REARM: Renewable energy based resilient deployment of Virtual Network Functions," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, Oulu, Finland, Jun. 2017, pp. 1–6.

[24] A. Gumaste, S. Sharma, T. Das, and A. Kushwaha, "Strategies for VNF placements in large provider networks," in *Proc. Opt. Fiber Commun. Conf. Exhibit. (OFC)*, Los Angeles, CA, USA, 2017, pp. 1–3.

[25] M. M. Tajiki, S. Salsano, L. Chiaraviglio, M. Shojafar, and B. Akbari, "Joint energy efficient and QoS-aware path allocation and VNF placement for service function chaining," *IEEE Trans. Netw. Service Manag.*, vol. 16, no. 1, pp. 374–388, Mar. 2019.

[26] M. Chen, S. C. Liew, Z. Shao, and C. Kai, "Markov approximation for combinatorial network optimization," *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6301–6327, Oct. 2013.

[27] S. Lee, W. Liu, and I. Hwang, "Markov chain approximation algorithm for event-based state estimation," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 3, pp. 1123–1130, May 2015.

[28] M. Chen, S. C. Liew, Z. Shao, and C. Kai, "Markov approximation for combinatorial network optimization," in *Proc. IEEE INFOCOM*, San Diego, CA, USA, Mar. 2010, pp. 1–9.

[29] Z. Xu, X. Zhang, S. Yu, and J. Zhang, "Energy-efficient virtual network function placement in telecom networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kansas City, MO, USA, May 2018, pp. 1–7.

WENCHEN HE was born in Laiwu, Shandong, China, in 1993. He received the B.S. degree from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2016. He is currently pursuing the Ph.D. degree with the Beijing University of Posts and Telecommunications. His research interests include intent-based network for the IoT and network function virtualization.

SHAOYONG GUO received the Ph.D. degree from the Beijing University of Posts and Telecommunication, Beijing, China, in 2013, where he is currently a Lecturer. His research interests include device management, the Internet of Things, ubiquitous networks, and smart grids.

YUN LIANG received the B.S. degree in electrical engineering from the China University of Mining and Technology, in 1999, and the M.S. degree in electrical engineering from the Southeast University, China, in 2002. He is currently a Senior Engineer with the Global Energy Interconnection Research Institute Co., Ltd., Beijing, China. His current research interests include communication networks and information systems in energy and power systems, especially the cyber-physical systems (CPS).

XUESONG QIU was born in 1973. He received the Ph.D. degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2000, where he is currently a Professor and Ph.D. Supervisor. Since 2013, he has been the Deputy Director of the State Key Laboratory of Networking and Switching Technology. He has authored more than 100 SCI/EI index papers. His current research interests include network management and service management. Currently, he presides over a series of key research projects on network and service management, including the projects supported by the National Natural Science Foundation and the National High-Tech Research and Development Program of China. His awards and honors include 13 national and provincial scientific and technical awards, including the national scientific and technical awards (second-class) twice.