IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON NEW WAVEFORM DESIGN AND AIR-INTERFACE FOR FUTURE HETEROGENEOUS NETWORK TOWARDS 5G

Received February 22, 2019, accepted March 7, 2019, date of publication April 12, 2019, date of current version April 30, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2909971

Computation Offloading Optimization Based on Probabilistic SFC for Mobile Online Gaming in Heterogeneous Network

HAO JIN, XIAOYING ZHU, AND CHENGLIN ZHAO

Key Laboratory of Universal Wireless Communications, Ministry of Education, Beijing University of Posts and Telecommunications, Beijing 100876, China

Corresponding author: Hao Jin (hjin@bupt.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61471062 and Grant 61431008, and in part by the State Major Science and Technology Special Projects under Grant 2017ZX03001014.

ABSTRACT To support new services targeted by 5G, great efforts have been taken not only on the research work of new waveform design and air interface but also on cloudification and softwarization for future heterogeneous network. As one of the most popular services toward 5G, cloud gaming offloads computation-intensive tasks to the cloud in order to alleviate the computation burden of mobile devices, but it introduces latency which deteriorates user experience especially for the delay-sensitive online game. In order to solve the optimization problem of resource allocation with the quality of experience guarantees and reduce the operational expenditures and capital expenditures of mobile operators for deploying online game, fog computing and network function virtualization are deemed as promising solutions. In this paper, a component-based approach is proposed to model online game based on the probabilistic service function chain. In order to obtain the optimal virtual function placement in the fog-enabled heterogeneous radio access network, the cost minimization of computation offloading on the data plane is formulated as an integer linear programming problem considering the constraints of application maximum tolerable latency, resource limitation, and user behavior. The optimization problem is NP-hard. To solve the problem with low complexity, the heuristic algorithm is proposed called Probabilistic Service function chain Embedding based on Cost Optimization(PSECO). The performances of the two algorithms are evaluated. The simulation results show that the costs are affected mainly by the number of components, the arrival rate of user requests, mobile user behavior, as well as the physical network topology and the number of users. The heuristic algorithm PSECO has optimal results with low complexity and it is suitable for large scale networks.

INDEX TERMS Computation offloading, fog computing, network function virtualization, online game, resource allocation, service function chain.

I. INTRODUCTION

Fifth generation (5G) radio access technology is expected to take a huge leap compared to the previous radio generations by supporting cognitive radio, machine type communication, the internet of things, besides traditional mobile broadband access. To support new services targeted by 5G, great efforts have been taken not only on the research work of new waveform design and air interface, but also on cloudification and softwarization for future heterogeneous network. As one of the most popular services towards 5G, cloud gaming provides gaming services to users with affordable, flexible and high performances, and it enables users to play game on thin clients by rendering everything in the cloud and simply streams the resulting high-quality video to users. However, the Quality of Experience (QoE) of cloud gaming suffers from high and unstable end-to-end delay due to the long transmission between users and the core network, which also causes bandwidth burden to the core network. To alleviate traffic burden to the core network as well as reduce delay and energy consumption of mobile devices, mobile online game is deployed in the fog computing environment [1]–[7].

Fog computing [8]–[10] was introduced by CISCO in 2012. It is an extension of cloud computing paradigm

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

from the core to the edge of the network, which is typically implemented in macro/small cell base stations (BSs), WiFi access points (APs), and so on [11]–[12]. By configuring computing-capable modules at these nodes, computation-intensive applications are able to be offloaded from the resource-constrained devices to the fog nodes and/or cloud nodes. Fog computing achieves a number of advantages including saving bandwidth resources, shortening latency, improving QoE of users [11] and reducing energy consumption of mobile devices which is especially beneficial for the Internet of Things [12]. However, the front-haul and backhaul bandwidth are occupied due to the frequently computation task offloading. Moreover, integrating new functions with dedicated hardware brings about more cost to operators [13]. Therefore, the operational expenditures (OPEX) and capital expenditures (CAPEX) of mobile operators are increased.

In order to reduce CAPEX and OPEX of the fog computing system for operators, network function virtualization (NFV) is regarded as a promising solution which makes flexible utilization of network and computation resources [8], [13]–[15]. NFV reduces OPEX and CAPEX by migrating Network Functions (NFs) from dedicated hardware appliances to software instances running on general purpose servers [14]. The NFs are partitioned as Virtualized Network Functions (VNFs). By composition of various VNFs, Service Function Chain (SFC) is formed, which is a network flow involving a set of VNFs interconnected by virtual links(VLs) as a chain [16].

Deploying a SFC mainly includes two processes, namely VNF instantiation and SFC mapping [17]. In order to obtain physical resources, VNFs need to be mapped to physical nodes, and virtual links among VNFs need to be mapped to physical links among physical nodes.

From the perspective of whether the executed VNF components are determined, SFCs are classified as deterministic SFCs and undeterminate SFCs. The deterministic SFCs mean that the requested VNFs are known in advance. These deterministic SFCs are mainly classified as linear SFCs [13], [14], [16], [18], parallel SFCs [18] and general SFCs according to the topology of SFCs composed by VNFs. General SFCs have not been investigated in detail yet in current research issues. The undeterminate SFCs are for uncertain requests, i.e. for the current function, the next function requested is uncertain [15].

SFC deployment is optimized based on different objectives to improve network resource utilization, reduce service delay as well as CAPEX and OPEX of mobile operators [13]–[19]. Online game is a typical case for NFV based SFC deployment in the fog computing environment.

Some research issues focusing on online game can be mainly classified as online game architecture, game application modelling and resource allocation, etc. [1]–[7], [20]–[27].

From the perspective of game architecture, fog gaming and cloud gaming are included, and cloud gaming is considered as the solution in most of the literatures [20]–[26]. In order to improve the performance of cloud gaming, fog/edge architecture is considered in [1]–[7]. For example, the hybrid edge-cloud architecture of cooperative edge servers placed nearby end-users to improve end-user coverage [2], the cloudlet-assisted network architecture in which the mobile devices are connected to the cloud server for real-time interactive game videos, while sharing the received video frames via an ad hoc cloudlet [3], and a lightweight system called CloudFog in which fog super-nodes cooperate to render game videos and stream them to their nearby players in order to reduce response time, bandwidth consumption and increase user coverage, while using the storage and computation resources on the cloud [5]. As another candidate solution instead of CloudFog, EdgeCloud [6] is deployed with a number of servers with specialized resources located near end-users, and these servers are responsible for hosting mobile online game including computing new game state and rendering game videos for players. In [7], the feasibility of edge computing for achieving satisfactory QoE is verified by using the open-source GamingAnywhere cloud gaming platform for mobile gaming.

From the perspective of game application modelling, literatures can be classified as two categories including session based modelling and component based modelling, which model call behavior among application software.

In the session based modelling, computation-intensive tasks are offloaded to resource-rich servers for computation, and the interaction of users with game servers is modelled by users sending action information/command continuously during online game [2], [5]–[7], [20]–[24].

In the component based modelling, mobile applications are decomposed of several loosely-coupled components so that applications can be quickly redeployed at runtime to offload parts of applications to other devices and make resource allocation more flexible [1], [28]–[30]. The call relationship among functional components in component based applications is modelled by a directed graph, which is classified as deterministic structure [1], [25], [27] and undeterministic structure. For component based computation tasks, the granularity of offloading is a key factor to affect the performance of the cloud/fog game [27], [28]. Principles of constructing the decomposed cloud gaming and research challenges from the perspective of decomposition granularity are presented in [27], and a trade-off between fine-grained and coarse-grained should be considered according to different demands [28]. In [1], a component based programming model is proposed and verified by dividing game logic into self-contained, remote executable as well as reusable software components. In [25], a component based game platform is designed to provide cognitive capabilities across the cloud gaming system, which supports click-and-play, intelligent resource allocation and partial offline execution.

As a case of the undeterministic structure, the IoT application for autonomous driving is considered whose SFC depends on the component execution [15]. The application is executed in sequence, in parallel, or by using more

complex substructures such as selections and loops which introduce uncertainty in the execution. However, the information related to user behavior is not considered in the application.

From the above modelling of online game, the interaction between users and game servers, as well as the computation task offloading are important behaviors to model. Modelling of frequent interactions with game servers mainly focuses on the control plane of gaming. Component based modelling contributes to the application programming model, and it enables modelling of the instance call behavior among different components including interactions both on the control plane and that on the data plane, which plays an important role in computation offloading.

Regarding online game, component instances called by mobile users depend on their skill(behavior) and the game component programming model, on the one hand, user call sequences of the application components are unknown, on the other hand, it is hard to know in advance how long the game would last, which component would be called next, as well as the amount of data to be offloaded [31]. In a word, due to the difficulties of modelling the undeterministic behavior of online game, the optimization of online game remains a challenge in the fog computing environment.

From the perspective of online game optimization, the resource allocation problem is addressed to optimize the overall game experience in [1]–[6], [20], [22], [23], [25]. Objectives are usually selected as latency minimization [1], [2], [20], cost minimization(CPU/GPU, energy, memory, etc.) [1], [23], network bandwidth consumption minimization [1], [3], [23], and profit maximization [22], etc.

To the best of our knowledge, no paper has focused on modelling the undeterministic online game in the cloud/fog based environment. In this paper, the online game of undeterministic structure is investigated in the fog computing environment on the basis of the component model. The probabilistic SFC is modelled based on interactive and uncertain features of online game, and the resource allocation problem is addressed with multi-objectives considering bandwidth, CPU/GPU resources and the number of VNF instances. The optimization problem is solved as a SFC mapping problem. The main contributions of the paper are as follows.

(1) To solve the optimization problem of resource allocation and reduce CAPEX and OPEX of mobile operators for online game, a component based approach is proposed based on probabilistic SFC in the fog computing environment, considering interactive and uncertain features of online game.

(2) In the scenarios of heterogeneous computing-enabled Radio Access Network(RAN) including MBS level and SBS level, the cost minimization of computation offloading on the data plane is formulated as an ILP problem by optimizing placement of VNFs considering the constraints of application maximum tolerable latency, resource limitation and user behavior.

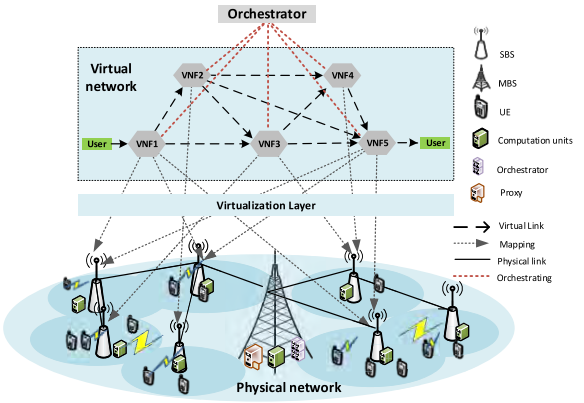

FIGURE 1. System model of online game in the fog-enabled heterogeneous RAN.

(3) A heuristic algorithm for Probabilistic SFC Embedding based on Cost Optimization (PSECO) is proposed to tackle the problem with low complexity.

(4) The ILP problem is solved and compared with the heuristic algorithm PSECO by simulation. Simulation results show that the costs are affected mainly by the number of components, the arrival rate of user requests, mobile user behavior as well as physical network topology and the number of users.

The rest of this paper is organized as follows. In Section II, the system model is given. In section III, the online game model is presented including system scenario, application model, probabilistic SFC model, computation model and communication model, and the problem is formulated. In section IV, a heuristic algorithm is presented to solve the optimization problem. In Section V, the ILP based optimization algorithm and the heuristic algorithm are evaluated. Finally, the paper is concluded in Section VI.

II. SYSTEM MODEL

Based on MANO [32], the system model of online game is illustrated in Fig.1. The online game is composed of various VNFs, and the user request oriented network flow is formed as SFC involving a set of VNFs on the virtual network.

In the fog-enabled heterogeneous RAN, several small base stations (SBSs) are located in the coverage of a macro base station (MBS). The SBSs and the MBS are enabled with small/medium-scale servers to provide computation and storage resources. That is to say, the BSs are fog nodes, and the online game VNF components are deployed in the fog layer. Either wired link or wireless link is deployed between BSs for communication. An orchestrator is resided on the MBS to support service orchestration, VNF orchestration and SFC mapping. The MBS also acts as the application proxy to process the service requests from mobile users.

The application proxy generates SFC by analyzing user information for the online game, and the orchestrator makes a decision on optimal placement for the SFC. According to the decision result, the SFC is mapped to the physical infrastructure. The SBSs provide computation resources for the

VNF components, and interact with mobile users via wireless links during online game. The physical links between SBSs are mapped as communication resources for the VLs on the virtual layer.

The detailed process is divided into two phases, namely SFC placement decision phase and computation offloading phase.

In the SFC placement decision phase, the interactions take place on the control plane. Firstly, the proxy located at the MBS receives service requests and generates a SFC for the online game according to user request information. Then, the proxy forwards the generated SFC to the orchestrator which performs the placement optimization algorithm based on the SFC. The optimization result depends on computation offloading optimization objectives and computation resource information of SBSs, etc. Based on the optimization result, the orchestrator sends the decision to those SBSs selected to instantiate VNFs, and responses to the proxy. Then the proxy sends the decision to the users. Finally, the VNFs of the online game are instantiated to serve mobile users.

In the computation offloading phase, user requests are satisfied by the VNF call sequences based on the application components. The user command information and state information are transmitted on the control plane, while the input/output arguments of VNF components are transmitted on the data plane. A VNF starts execution after receiving input arguments and user command information. When a VNF is completed, it returns state information to the user, the user sends a service request (command) for the next VNF on the control plane, which is a VNF call sent to the next VNF with the output arguments of the current VNF. The calls between VNFs are executed according to the offloading decision, and the input/output arguments are transmitted among the related SBSs to which the VNFs are offloaded. When the last VNF is completed, it returns results to the user on the data plane. During the time when mobile users are playing, the VNFs interact with users on the control plane if necessary.

An example of the online game provision procedure based on VNF chain is shown in Fig.2, in which the user requests VNF1 and VNF3, while VNF1 is decided to be placed on SBS2, and VNF3 to be placed on SBS3 according to the orchestration.

From the above behavior modelling of online game, it indicates that the VNFs called by mobile users are undeterministic, which is a challenge for the SFC embedding. Furthermore, the traffic between VNFs should be considered including interactions on the control plane and that on the data plane. Since the amount of data transmitted on the control plane is small compared with that transmitted on the data plane, the modelling of the data plane is addressed, which is presented in section III.

III. MODELING OF ONLINE GAME IN THE FOG-ENABLED HETEROGENEOUS RAN

In this section, probabilistic SFC model is proposed based on component based online game, and embedded into the fog-based heterogeneous RAN. The section includes system scenario, application model based on components, probabilistic SFC model based on VNF components, computation model and communication model. Based on the modelling, the problem is formulated as an ILP to minimize the cost of deploying SFC by mapping the probabilistic SFC into physical infrastructure considering the constraints of application maximum tolerable latency, resource limitation and user behavior.

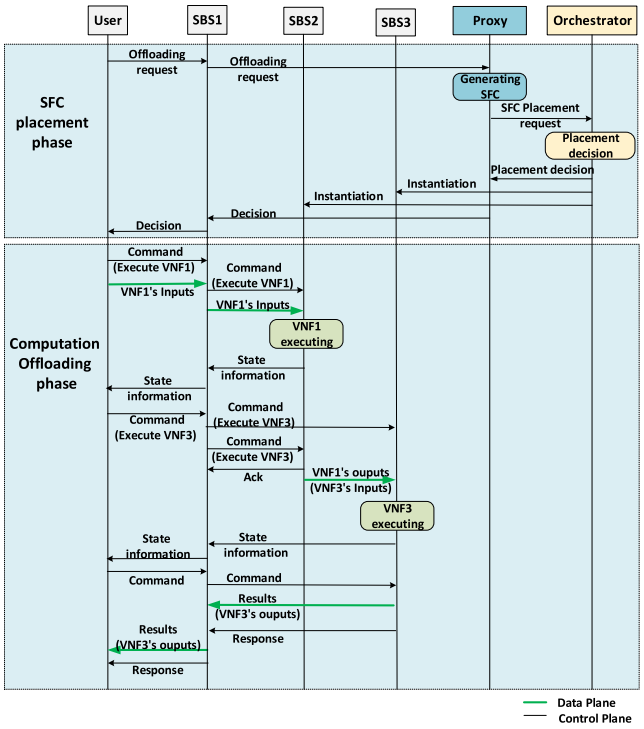

FIGURE 2. Online game provision procedure based on SFC embedding.

A. SYSTEM SCENARIO

As shown in Fig.1, assume that there is a MBS, which acts as an orchestrator and an application proxy. In the coverage of the MBS, there are N_{SBS} SBSs to provide computation resources for online game in terms of the number of CPUs/GPUs. A two-dimensional grid topology is used to model the distribution of SBSs in Fig.3 [33]. The networking topology of SBSs is denoted as $G_{phy}(V_{SBS}, E_{SBS})$, where V_{SBS} represents the set of SBSs, and E_{SBS} represents the set of physical links between SBSs. v_k denotes the SBSs in tier k, and the number of SBSs in tier k is $4 * k$. The topology is composed of up to K tiers of SBSs. The SBSs are connected by wired links, and the average degree of the topology is defined as av_d. In the coverage area, N_{UE} users are uniformly distributed with different online game requests.

B. APPLICATION MODEL BASED ON VNF COMPONENTS

1) GRAPH MODEL BASED ON VNF COMPONENTS

In the application layer, the online game is divided into several VNF components, and the behavior of the application execution is modelled as calls between components,

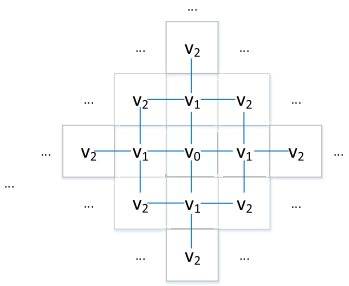

FIGURE 3. Networking model of SBSs.

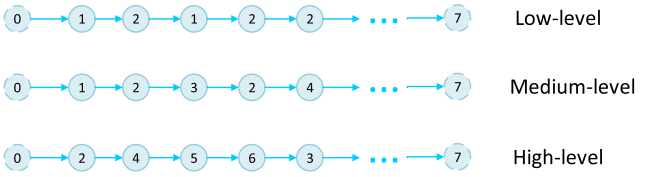

FIGURE 4. An example of component call sequences for different user levels.

therefore, the call sequences based on these VNF components are formed as SFCs according to the service requests, which reflect the user behavior during the game.

A directed graph $G_{call}(V_{call}, E_{call})$ is defined to represent the call relations among VNF components of the online game. The vertex set V_{call} denotes all components of the application, where a vertex represents a VNF component. The edge set E_{call} represents the call relations among VNF components. Suppose the online game is composed of n components, then, the set V_{call} is denoted by $V_{call} = \{0, 1, 2, \ldots, n, n+1\}$, where VNF component 0 and component $n+1$ do not require resources and are placed at the SBSs to which the users are attached, namely representing the source and destination of the application based on the request. The edge $e_{call}^{i,j} \in E_{call}$ denotes that component i calls j, which means that j is the next component called by component i. The VNF component starts execution when it receives the input data from its previous VNF component.

2) MODELING OF USER BEHAVIOR BASED ON VNF COMPONENT GRAPH

Different experiences are provided to online game users during the game, which mainly depend on the design of the game programming. However, the primary experiences are usually the level and the time allowed to play the game, which are configured according to the skill(behavior) of users as well as the grade of difficulty. For example, the longer the game lasts, the higher the level is, and more difficult the game is.

Considering the component based application model and user behavior(skill), we assume that the game is divided into three levels, namely low-level, medium-level and high-level to differentiate user levels(behavior). Supposing that the level is related to the VNF components, that is to say, the level corresponds to the components which are called. Low-level users only call some of the VNF components, medium-level users call more of the VNF components than that of low-level users, while high-level users call all VNF components arbitrarily. When mobile users play online game, different VNF component call sequences are generated depending on their levels(behavior). For example, users can call VNF components arbitrarily and repeatedly within the set of the components allowed by their levels.

Assume that user levels do not change during the SFC embedding and computation offloading. For user u, the level is assumed to be $level(u)$ and the length of the call sequence is $length(u)$. Thus, user u generates a VNF component call sequence $s(u)$ within the set of the components allowed by its level. An example of the VNF component call sequences for different user levels is shown in Fig.4 when the game application has six VNF components ($n = 6$). In Fig.4, users can call a VNF component repeatedly, and the length of the application sequences is undeterministic when they play the game.

Assume that service requests follow Poisson process with an average arrival rate of λ requests per user. The request argument of user u is denoted as $R(u) = \{DS(u), T_{max}(u)\}$, where $DS(u)$ denotes the load of data offloaded by user u, and $T_{max}(u)$ is the maximum tolerable latency of the request required by user u. The load of the total requests for users is expressed as

$$l_{total} = \sum_{u=1}^{N_{UE}} DS(u) \tag{1}$$

Based on the application model and user behavior, the probabilistic SFC is modelled to illustrate the undeterministic call sequences in section III-C.

C. PROBABILISTIC SFC MODEL BASED ON VNF COMPONENT GRAPH

For the VNF component based online game, the SFC is undeterministic since the next calling VNF depends on the user behavior. In this section, the probabilistic SFC is modelled based on Markov Process.

The components contained in the online game constitute a state space, where state i indicates that component i is called. The state transition probability matrix $\mathbf{P_{user}(u)}$ is defined for user u between components, where the element of the matrix $P_{user}^{i,j}(u)$ is the transition probability of user u from component i to j. The transition probability of users is computed and recorded at regular intervals by the application proxy. Let $s(u)$ represents the call sequence of user u playing the game, a new $s(u)$ causes a change in $\mathbf{P_{user}(u)}$, and $\mathbf{P_{user}(u)}$ represents the information of user u to play the game.

In order to illustrate call behavior of all users requesting the game, let $\mathbf{P_{app}}$ be the state transition probability matrix for the application between components, which is composed of $P_{app}^{i,j}$ indicating the state transition probability from component i to j of the application, and $P_{app}^{i,j} = \frac{1}{N_{UE}} * \sum_{u=1}^{N_{UE}} P_{user}^{i,j}(u)$. $\mathbf{P_{app}}$

is computed by the application proxy, which is related to the number of users and user behavior.

Denote $G_{SFC}(V_{SFC}, E_{SFC})$ as the graph to represent the probabilistic SFC. The vertex set V_{SFC} represents the VNFs that the SFC has, namely $V_{SFC} = \{VNF_i | i = 0, 1, 2, \ldots, n, n+1\}$, where 0 and $n+1$ just indicates the first and last VNF placed on the SBSs attached by users and they do not consume resources. The set E_{SFC} represents the probabilistic calls of VNFs. The weight of edge $e^{i,j}_{SFC} \in E_{SFC}$ is $P^{i,j}_{app}$, which is the probability of selecting j after i. The sum of weights of all outgoing edges of a vertex is 1, namely,

$$\sum_{j=0}^{n+1} P^{i,j}_{app} = 1, \quad \forall i \in \{0, 1, 2, \ldots, n, n+1\} \tag{2}$$

The state transition graph based on Markov model for the probabilistic SFC is shown in Fig.5.

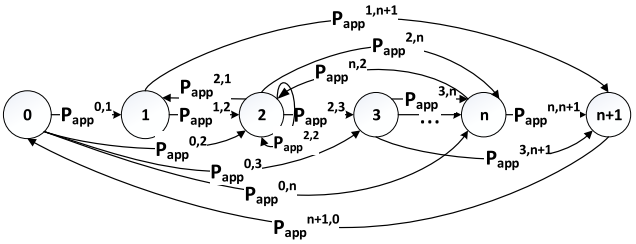

FIGURE 5. State transition graph for the probabilistic SFC.

For the application, the embedded SFCs are different depending on the $\mathbf{P_{app}}$. For simplicity, assuming that the input and output data augments of VNFs are the same. According to Markov model, the steady-state probability distribution is defined as $\Pi = \{\pi_i | i = 0, 1, 2, \ldots, n, n+1\}$. The formula for solving the steady-state probability is as follows [33]:

$$\Pi * \mathbf{P_{app}} = \Pi \tag{3}$$

$$\sum_{i=0}^{n+1} \pi_i = 1 \tag{4}$$

where π_i represents the steady-state probability of requesting VNF_i.

D. COMPUTATION MODEL IN THE FOG-ENABLED HETEROGENEOUS RAN

In the fog-enabled heterogeneous RAN, the SBSs provide computation resources for the VNF components during online game, and the computation resources are modelled by available CPUs/GPUs of SBSs. The VNFs are deployed on the SBSs, and each VNF has a predetermined processing capacity.

l_{VNF_i} is defined as the computation load to the VNF_i, then it can be obtained as:

$$l_{VNF_i} = l_{total} * \pi_i \tag{5}$$

Since each *VNF* has a predetermined processing capacity, when the load to VNF_i exceed the maximum processing capacity, the orchestrator deploys another VNF_i instance to process the load.

In order to model the call behavior among different instances, *Ins* is defined as the set of instances, and $ins_i^{f_i}, f_i \in \{1, 2, \ldots I_i\}$ represents the instance f_i of VNF_i, where I_i is the number of instances of VNF_i. The total number of instances for VNFs is N_{ins}. $l_{ins_i^{f_i}}$ is denoted as the computation load to $ins_i^{f_i}$. For $ins_i^{f_i}$, $R_i^{f_i}$ represents the number of CPUs/GPUs required to process unit load, and $R_0^{f_0} = 0, R_{n+1}^{f_{n+1}} = 0$. $Cap_i^{f_i}$ represents the processing capacity of $ins_i^{f_i}$. $ins_0^{f_0}$ and $ins_{n+1}^{f_{n+1}}$ are placed at the corresponding SBSs to which the users are attached, indicating the start and end of the application, respectively.

Assume that the available CPUs/GPUs of SBS m is $A_{com}(m)$, and the cost of unit resource is $C_{com}(m)$. The instantiation cost of $ins_i^{f_i}$ at SBS m is defined as $C_{ins_i^{f_i}}(m)$. $T_{pins_i^{f_i}}(m)$ represents the processing delay of $ins_i^{f_i}$ on SBS m for unit load. Then the computation node cost can be formulated by CPU/GPU cost and instantiation cost, and the delay can be formulated by the processing delay.

E. COMMUNICATION MODEL IN THE FOG-ENABLED HETEROGENEOUS RAN

Virtual links are interconnected between instances on the virtual layer, representing traffic load transmission between VNF instances. An example of traffic load distribution among instances is shown in Fig.6, where $\delta_i^{f_i}$ is defined as the ratio of the load to instance f_i of VNF_i to that to VNF_i,

$$\delta_i^{f_i} = \frac{l_{ins_i^{f_i}}}{l_{VNF_i}} \tag{6}$$

The load from instance $ins_i^{f_i}$ to instance $ins_j^{f_j}$ is $l_{ins_i^{f_i}} * \delta_j^{f_j} * P^{i,j}_{app}$.

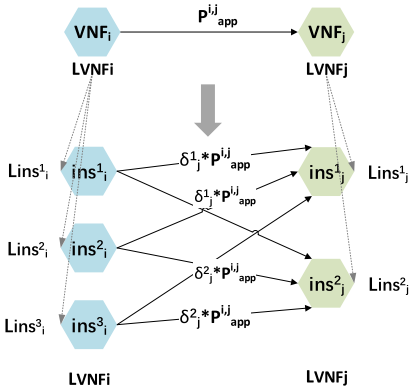

FIGURE 6. An example of traffic load distribution among instances on the virtual network.

Virtual links on the virtual network are mapped to physical links on the physical network to obtain the communication resources, which leads to communication cost including wired cost and wireless cost. Let $C_{wirless}$ denote the cost of wireless bandwidth for unit load. The achievable

uplink transmission rate r_{up} from users to SBSs is given as $r_{up} = W\log_2(1+\frac{g*P_t}{\sigma^2})$. Define r_{down} as the achievable downlink rate from SBSs to users. Assume that SBSs communicate with each other via wired links. The available capacity of the bandwidth between SBS m and SBS p is denoted as $A_{wired}(m,p)$, and the cost of the bandwidth for unit load is $C_{wired}(m,p)$. $H(m,p)$ represents the number of hops between SBS m and SBS p. Communication between physical nodes to which VNFs are mapped causes transmission delay. Define $T_{tr}(m,p)$ as the delay for unit load between SBS m and SBS p.

Based on the component based game model, computation model and communication model in fog computing environment, the problem is formulated aiming at minimizing cost for online game in section III-F.

F. PROBLEM FORMULATION

The resource allocation optimization problem of computation offloading for online game in the fog-enabled heterogeneous radio access network is formulated as an ILP with multi-objectives aiming at minimizing the total cost including CPU/GPU cost, instantiation cost and transmission cost, with Service Delivery Time(SDT) [34] as one of the constraints. The variables used in the formulation are given in Table 1.

According to the modelling, the cost in terms of the CPU/GPU is computed in (7),

$$Cost_{cpu} = \sum_{i\in V_{SFC}}\sum_{f_i\in I_i}\sum_{m\in V_{SBS}} l_{ins_i^{f_i}} * R_i^{f_i} * x_m^{i,f_i} * C_{com}(m) \quad (7)$$

The VNF instantiated cost can be obtained in (8),

$$Cost_{ins} = \sum_{i\in V_{SFC}}\sum_{f_i\in I_i}\sum_{m\in V_{SBS}} x_m^{i,f_i} * C_{ins_i^{f_i}}(m) \quad (8)$$

The sum of CPU/GPU cost and instantiation cost is defined as the Node cost.

The transmission cost includes wireless transmission cost between users and SBSs, and wired transmission cost among SBSs. That is to say, $Cost_{Trans} = Cost_{wired} + Cost_{wireless}$.

The wired transmission cost is got as in (9),

$$\begin{aligned} IR_{m,p}^{(i,f_i),(j,f_j)} &= l_{ins_i^{f_i}} * P_{app}^{i,j} * \delta_j^{f_j} * y_{m,p}^{(i,f_i),(j,f_j)} \\ IW_{wired_{cost}}^{m,p} &= IR_{m,p}^{(i,f_i),(j,f_j)} * H(m,p) * C_{wired}(m,p) \\ Cost_{wired} &= \sum_{i\in V_{SFC}}\sum_{f_i\in I_i}\sum_{m\in V_{SBS}}\sum_{j\in V_{SFC}}\sum_{f_j\in I_j}\sum_{p\in V_{SBS}} IW_{wired_{cost}}^{m,p} \end{aligned} \quad (9)$$

The wireless transmission cost is computed as in (10),

$$Cost_{wireless} = (l_{total} + l_{VNF_{n+1}}) * C_{wireless} \quad (10)$$

where $l_{VNF_{n+1}}$ is defined as the load to VNF_{n+1}.

Therefore, the total cost is formulated as in (11),

$$TotalCost = Cost_{cpu} + Cost_{ins} + Cost_{wired} + Cost_{wireless} \quad (11)$$

TABLE 1. Variables used in the formulation.

Notations	Description
V_{SBS}	Set of SBSs covered by the MBS
V_{SFC}	Set of VNFs for the SFC of online game
I_i	Number of instances for VNF_i
N_{ins}	Number of instances required to serve users
l_{total}	Load of total requests for users
l_{VNF_i}	Load to VNF_i
$ins_i^{f_i}$	Instance f_i of VNF_i
$l_{ins_i^{f_i}}$	Load to the instance f_i of VNF_i
$P_{app}^{i,j}$	Transition probability between VNF_i and VNF_j
$R_i^{f_i}$	Required CPUs/GPUs of instance f_i of VNF_i per unit load
$Cap_i^{f_i}$	Processing capacity of instance f_i of VNF_i
$T_{tr}(m,p)$	Transmission delay for unit load between SBS m and SBS p
$T_{pins_i^{f_i}}(m)$	Processing delay of instance f_i of VNF_i on SBS m per unit load
$T_{max}(u)$	Maximum tolerable latency for user u
$A_{com}(m)$	Available CPUs/GPUs of SBS m
$A_{wired}(m,p)$	Available capacity of the wired bandwidth between SBS m and SBS p
$C_{com}(m)$	Cost of unit CPU/GPU for SBS m per unit load
$C_{wireless}$	Cost of wireless bandwidth per unit load
$C_{wired}(m,p)$	Cost of wired bandwidth between SBS m and SBS p per unit load
$C_{ins_i^{f_i}}(m)$	Instantiation cost of instance f_i of VNF_i at SBS m
r_{up}	Uplink transmission rate from users to SBSs
r_{down}	Downlink transmission rate from SBSs to users
$\delta_i^{f_i}$	Ratio of the load to instance f_i of VNF_i to the load to VNF_i
$H(m,p)$	Hops between SBS m and SBS p
x_m^{i,f_i}	Binary placement decision for instance f_i of VNF_i mapped to SBS m
$y_{m,p}^{(i,f_i),(j,f_j)}$	Binary placement decision for instance f_i of VNF_i mapped to SBS m, and instance f_j of VNF_j mapped to SBS p

where x_m^{i,f_i} is a binary variable which represents that whether $ins_i^{f_i}$ is processed at SBS m. If the $ins_i^{f_i}$ is processed at SBS m, $x_m^{i,f_i} = 1$, otherwise, $x_m^{i,f_i} = 0$. $y_{m,p}^{(i,f_i),(j,f_j)}$ is a binary variable representing that whether the virtual link between $ins_i^{f_i}$ and $ins_j^{f_j}$ is hosted by the physical link between SBS m and SBS p. If the virtual link between $ins_i^{f_i}$ and $ins_j^{f_j}$ is hosted by the physical link between SBS m and SBS p, $y_{m,p}^{(i,f_i),(j,f_j)} = 1$, otherwise, $y_{m,p}^{(i,f_i),(j,f_j)} = 0$.

In order to consider the user experience delay as one of the constraints for the formulation, SDT is computed. The SDT is usually used to measure the user experience delay, which is defined as the total time for the requests to reach the server, being processed and reach back the terminal [34]. According to the definition, the SDT of online game in our paper includes the average transmission time of the input arguments to VNFs and the average time being processed

per interaction (note that the delay on the control plane is not considered).

The average processing delay of VNFs is got as,

$$AV_{Tnode} = \frac{1}{N_{ins}} * (\sum_{i \in V_{SFC}} \sum_{f_i \in I_i} \sum_{m \in V_{SBS}} l_{ins_i^{f_i}} * T_{pins_i^{f_i}}(m) * x_m^{i,f_i}) \tag{12}$$

The total delay of wired bandwidth transmission between SBSs can be obtained as in (13),

$$\begin{aligned} IE_{wired_{time}}^{m,p} &= IR_{m,p}^{(i,f_i),(j,f_j)} * H(m,p) * T_{tr}(m,p) \\ T_{wired} &= \sum_{i \in V_{SFC}} \sum_{f_i \in I_i} \sum_{m \in V_{SBS}} \sum_{j \in V_{SFC}} \sum_{f_j \in I_j} \sum_{p \in V_{SBS}} IE_{wired_{time}}^{m,p} \end{aligned} \tag{13}$$

The wireless transmission delay between users and SBSs is got in (14),

$$T_{wireless} = l_{total}/r_{up} + l_{VNF_{n+1}}/r_{down} \tag{14}$$

Therefore, the average transmission delay for each link is,

$$AV_{Ttrans} = \frac{1}{num_{VL} + 2} * (T_{wired} + T_{wireless}) \tag{15}$$

where num_{VL} represents the number of VLs between instances that need to be mapped to the physical links.

Therefore, the average experience delay of users can be obtained as (16),

$$AV_{time} = AV_{Tnode} + AV_{Ttrans} \tag{16}$$

Finally, the resource allocation optimization problem of computation offloading can be formulated as:

$$\begin{aligned} &min \quad TotalCost \\ &s.t.\ C1: \ AV_{time} \le \min(T_{max}(u)) \\ &\quad C2: \ \sum_{m \in V_{SBS}} x_m^{i,f_i} = 1, \quad \forall i \in V_{SFC},\ f_i \in I_i \\ &\quad C3: \ \sum_{i \in V_{SFC}} \sum_{f_i \in I_i} l_{ins_i^{f_i}} * R_i^{f_i} * x_m^{i,f_i} < A_{com}(m), \quad \forall m \\ &\quad C4: \ \sum_{i \in V_{SFC}} \sum_{f_i \in I_i} \sum_{j \in V_{SFC}} \sum_{f_j \in I_j} IR_{m,p}^{(i,f_i),(j,f_j)} \\ &\qquad < A_{wired}(m,p), \quad \forall m,\ p \in V_{SBS} \\ &\quad C5: \ l_{ins_i^{f_i}} \le Cap_i^{f_i}, \quad \forall i \in V_{SFC},\ f_i \in I_i \\ &\quad C6: \ \sum_{p \in V_{SBS}} y_{m,p}^{(i,f_i),(j,f_j)} - \sum_{p \in V_{SBS}} y_{p,m}^{(i,f_i),(j,f_j)} \\ &\qquad = x_m^{i,f_i} - x_m^{j,f_j}, \quad \forall m \in V_{SBS},\ \forall i,\ f_i,\ j,\ f_j \end{aligned} \tag{17}$$

Constraint C1 indicates that the user experience delay should be less than or equal to the maximum tolerable delay. Constraint C2 indicates that a VNF instance can be mapped only once. Constraint C3 limits the number of CPUs/GPUs provided by the SBS less than or equal to its capacity. Constraint C4 limits the bandwidth provided by the wired link not larger than its capacity. Constraint C5 indicates that the traffic handled by each VNF instance should not exceed its maximum processing capacity. Finally, the C6 constraint ensures the network flow conservation.

The optimization problem is a general integer linear programming (ILP), which is NP-hard. For real-time applications and large scale networks, it is difficult to find an optimal solution in an efficient way. To solve the problem with low complexity, a heuristic algorithm is proposed in section IV.

IV. SOLUTION TO THE OPTIMIZATION OF PROBABILISTIC SFC EMBEDDING

In this section, a heuristic algorithm is proposed called Probabilistic SFC Embedding based on Cost Optimization (PSECO).

The heuristic algorithm PSECO contains two steps: the first step is to map the virtual nodes of instances to the physical network, and the second step is to map the virtual links among instances to the physical links based on the shortest path. If the output decision cannot meet the resource constraints, a virtual node on the physical node with the most occupied resources is migrated to the physical node with more resources.

The algorithm is depicted in Algorithm 1.

In the algorithm, $IW_{wired_{cost}}^{p,m}$ represents the wired cost from $ins_j^{f_j}$ to $ins_i^{f_i}$, in which $ins_j^{f_j}$ is mapped to SBS p, and $ins_i^{f_i}$ is processed at SBS m. Define $V(ins_i^{f_i}, m)$ as the node cost for $ins_i^{f_i}$ placed at SBS m. The output $\mathbf{X}$ is a decision matrix, where

$$X_m^{i,f_i} = \begin{cases} 1 & ins_i^{f_i} \text{ is placed at SBS } m \\ 0 & ins_i^{f_i} \text{ is not placed at SBS } m \end{cases} \tag{18}$$

Based on the ILP optimization algorithm and the heuristic algorithm PSECO, the performance of them is evaluated in section V.

V. PERFORMANCE EVALUATIONS

In this section, the performances of the exact ILP solution(we call it ILP based optimization algorithm later) and the heuristic algorithm PSECO are evaluated with various parameters including the number of components of online game n, the arrival rate of user requests λ, the ratio of high-level users to total users R_h, the number of tiers K, the average degree of the physical network av_d, and the number of users N_{UE}.

The performance metrics in the simulation include Node cost, Wired cost, Wireless cost and Total cost. Node cost is the sum of CPU/GPU cost and instantiation cost. The instantiation cost depends on the number of instances [15].

The performances of the online game computation offloading optimization algorithms are evaluated by MATLAB with Monte Carlo method. The ILP based optimization algorithm is implemented using *Gurobi* optimizer [35]. The results are averaged from 1000 simulations.

The detailed simulation parameter settings are given in V-A, and the evaluation results are presented in V-B. The complexity of the algorithms is analyzed in V-C.

Algorithm 1 PSECO

Input: $V_{SFC}, E_{SFC}, V_{SBS}, E_{SBS}$
Calculate steady-state load to instances;
Calculate steady-state load transferred between instances;
for *ins_i* in *Ins* **and** *ins_i* is no.mapped **do**
  for each physical node $m \in V_{SBS}$ **do**
    for all $ins\_j \in Ins$ whose traffic is transferred to *ins_i* **do**
      if no.mapped **then**
        $value \leftarrow +\infty$;
        for each physical node $p \in V_{SBS}$ **do**
          $value \leftarrow min(value, IW^{p,m}_{wired_{cost}})$;
        end for;
      else
        $loc\_j \leftarrow$ physical node embedded by *ins_j*;
        $value \leftarrow IW^{p,m}_{wired_{cost}}$;
      end if;
      $embedvalue(m) \leftarrow embedvalue(m) + value$;
    end for;
    $embedvalue(m) \leftarrow embedvalue(m) + V(ins\_i, m)$;
  end for;
  $loc\_i \leftarrow \arg\min(embedvalue)$;
  update **X**;
end for;
while (**X** not meeting resource constraints) **and** ($num < num\_loop$)
  migrate the most resource-intensive instance of the most resource-consuming physical node to the physical node with more resources;
end while;
Assign physical links for the VLs;
Output: X

A. SIMULATION PARAMETERS

The simulation parameters are given in this section. In the simulation, if it is not clearly stated, assuming that the number of medium-level users and the number of low-level users are equally divided. The uncertainty on the VNF resource demands is considered, since the CPU/GPU utilization varies according to the VNF processing load. The arrival rate of requests follows Poisson distribution. The size of the offloaded data changes. Assume that the number of requests fluctuates with a maximum deviation a under a nominal value μ [36].

Since the cost of each unit computation and bandwidth resource depends on various factors including practical operation and management experience of operators, for simplicity, C_{com} and C_{wired} are assigned as 1 according to [19]. Similarly, the instantiation cost for VNFs and the wireless cost for unit load are also set as 1.

The detailed parameters are listed in Table 2, including the server capacity [23], maximum tolerable latency [23], [26] and fog-enabled RAN parameters [37], [38]. The ER model is used as the network topology model for the physical network on the SBS level.

TABLE 2. Parameters used in the formulation.

Notations	Value
The maximum tolerable latency(T_{max})	200ms[23,26]
The number of CPU cores of SBSs(A_{com})	4[23]
Transmit power of users(P_t)	0.01W [38]
Transmit power of SBSs	0.1W [38]
Pass loss exponent(ς)	4
Background noise(σ^2)	$-100dbm$ [37]
Bandwidth of sub-channel (W)	5MHz
Coverage of of SBSs	50m

B. EVALUATION RESULTS AND DISCUSSIONS

In this subsection, the impact of the parameters to the cost is investigated including the number of components of online game n, the arrival rate of user requests λ, the ratio of high-level users to total users R_h, the number of tiers K, the average degree of the physical network on the SBS level av_d, as well as the number of users N_{UE}.

1) IMPACT OF THE NUMBER OF COMPONENTS TO THE COST

Fig.7 shows the impact of the number of components n to the cost when $K = 1$, $N_{UE} = 100$, $R_hx = 0.5$, $\lambda = 3$ and $av_d = 3$. The cost includes total cost, node cost, wired cost and wireless cost.

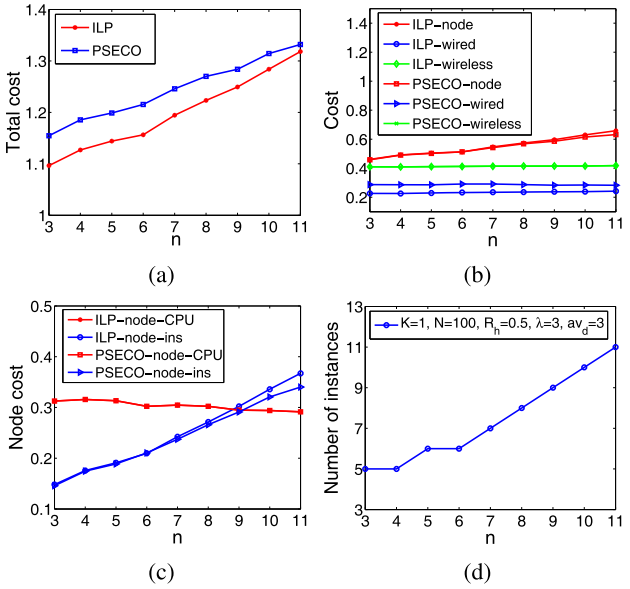

FIGURE 7. Impact of n to (a) total cost, (b) different cost, (c) node cost and (d) the number of instances.

In Fig. 7(a), the ILP based optimization algorithm is more efficient than PSECO as expected in terms of the total cost. When n increases, it indicates that the total cost is gradually

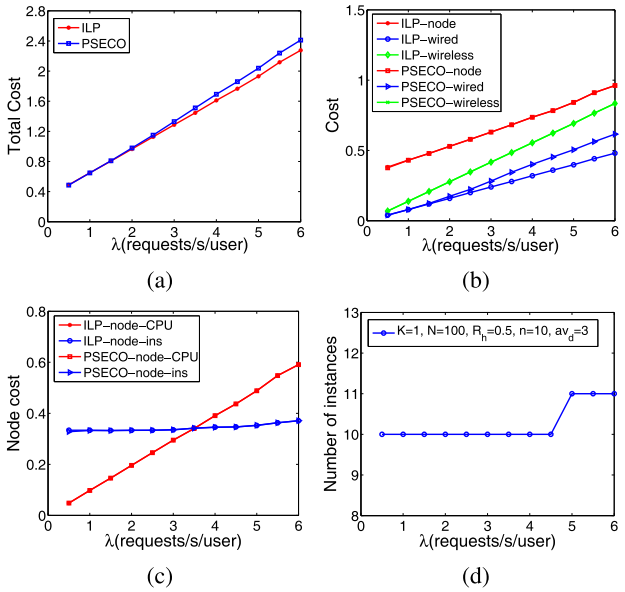

FIGURE 8. Impact of λ to (a) total cost, (b) different cost, (c) node cost and (d) the number of instances.

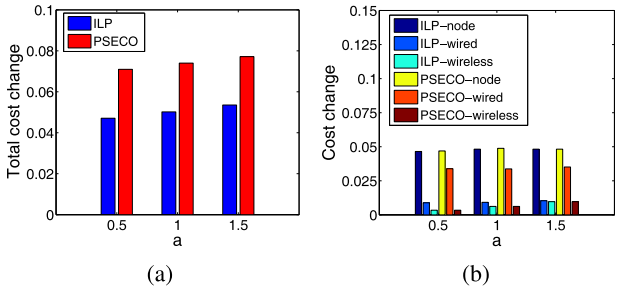

FIGURE 9. Impact of *a* to (a) total cost change and (b) cost change.

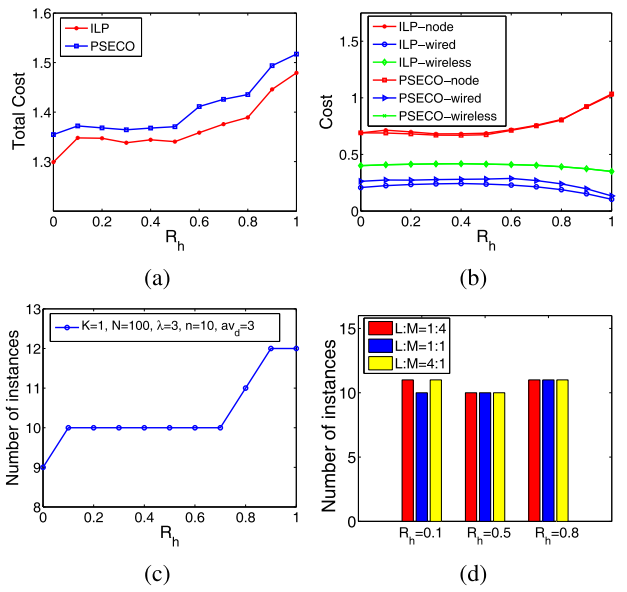

FIGURE 10. Impact of R_h to (a) total cost, (b) different cost, (c) the number of instances when the proportion of low-level and medium-level users is 1:1 and (d) the number of instances when L:M changes.

increasing. In Fig. 7(b), the node cost changes most obviously among the three costs, and the other two costs remain nearly the same. Since the node cost is divided into CPU/GPU cost and instantiation cost, as can be seen from Fig. 7(c) and Fig. 7(d), when n increases, the node cost becomes higher due to the increase of the number of instances.

2) IMPACT OF THE ARRIVAL RATE OF USER REQUESTS TO THE COST

Fig.8 shows the impact of the arrival rate of user requests λ to the cost when $K = 1$, $N_{UE} = 100$, $R_h = 0.5$, $\lambda = 3$ and $n = 10$.

In Fig. 8(a), the ILP based optimization algorithm is more efficient than PSECO, but the difference is small. When λ increases, it reveals that the total cost becomes higher, and the three costs also become larger in Fig. 8 (b). With the same number of components n, as λ increases, the instantiated cost becomes higher in Fig. 8(c), because the number of instances increases to meet more user requests. The change of the number of instances to the arrival rate of user requests is shown in Fig. 8(d).

Fig.9 shows the impact to the cost change when the number of requests has a fluctuation with a nominal value of 3 and a maximum deviation of a. The cost change is defined as the difference between the cost when the arrival rate fluctuates ($a > 0$) and the cost when $a = 0$. From Fig. 9(a), although a increases, the total cost changes slightly. The total cost change ranges from 0.04 to 0.08, and it is mainly caused by the change of node cost and wired cost (Fig. 9(b)).

3) IMPACT OF MOBILE USER BEHAVIOR TO THE COST

Fig.10 shows the impact of the ratio of high-level users to total number of users R_h to the cost when $av_d = 3$, $K = 1$, $N_{UE} = 100$, $\lambda = 3$ and $n = 10$.

The ILP based optimization algorithm is more efficient than PSECO in Fig. 10(a), when R_h increases, the total cost increases as well. From Fig. 10(b), since the set of components that high-level users call is large, with the increase of R_h, the number of instances required becomes larger, which causes the increase of the node cost. In Fig. 10(b) and Fig. 10(c), when R_h is bigger than 0.5, it reveals that the node cost and the number of instances change rapidly. Fig. 10(d) shows the impact of R_h to the number of instances when changing the proportion of low-level and medium-level users (L: M).

4) IMPACT OF THE NUMBER OF TIERS TO THE COST

Fig.11 shows the impact of the number of tiers K when $av_d = 3$, $N_{UE} = 100$, $R_h = 0.5$, $\lambda = 3$ and $n = 10$.

Fig. 11(a) and Fig. 11(b) show the impact of K to total cost and different cost, respectively. It can be seen from Fig. 11(a) that the ILP based optimization algorithm has a lower total cost than that of PSECO, and as K increases, the total cost increases. Fig. 11(b) shows the impact of K on various costs. It indicates that the wired cost of PSECO

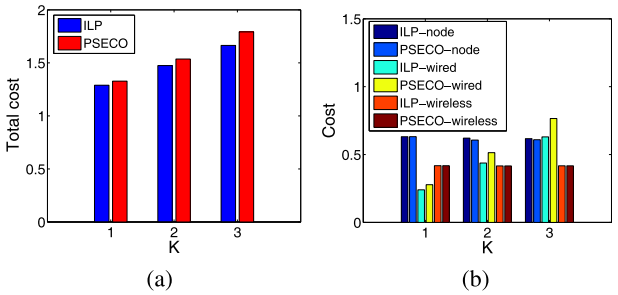

FIGURE 11. Impact of K to (a) total cost and (b) different cost.

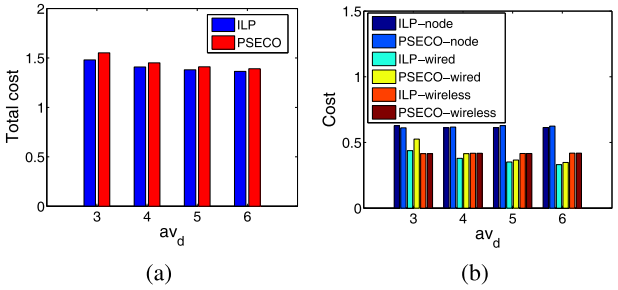

FIGURE 12. Impact of av_d to (a) total cost and (b) different cost.

is better than that of the ILP based optimization algorithm, while the node cost of PSECO is a little lower than that of the ILP based optimization algorithm. As K increases, only the wired bandwidth cost increases, and the other costs remain nearly the same.

5) IMPACT OF THE AVERAGE DEGREE OF THE NETWORK TO THE COST

The average degree of the network on the SBS level affects the cost of computation offloading. Fig.12 shows the impact of the average degree of the network av_d to the cost when $N_{UE} = 100$, $R_h = 0.5$, $\lambda = 3$, $K = 2$ and $n = 10$.

In Fig. 12(a) and Fig. 12(b), the impact of av_d to total cost and different cost is shown. It reveals that when av_d increases, the total cost is reduced, this is because more and more SBSs are connected directly as av_d increases, and the bandwidth cost caused by multi hop transmission is replaced by that of one hop transmission.

6) IMPACT OF THE NUMBER OF USERS TO THE COST

Fig.13 shows the impact of the number of users N_{UE} when $av_d = 3$, $K = 1$, $R_h = 0.5$, $\lambda = 5$ and $n = 10$.

The impact of N_{UE} to the cost is given in Fig.13. It indicates that the ILP based optimization algorithm has a lower total cost than that of PSECO. In Fig. 13 (a) and Fig. 13(b), when N_{UE} increases, each cost becomes higher. For the node cost, the CPU/GPU cost increases rapidly with N_{UE} and the instantiation cost increases slowly in Fig. 13(c). Due to the increase of the number of users, the CPU/GPU cost is larger because of more traffic load from users. The increase of the number of users also brings about more number of

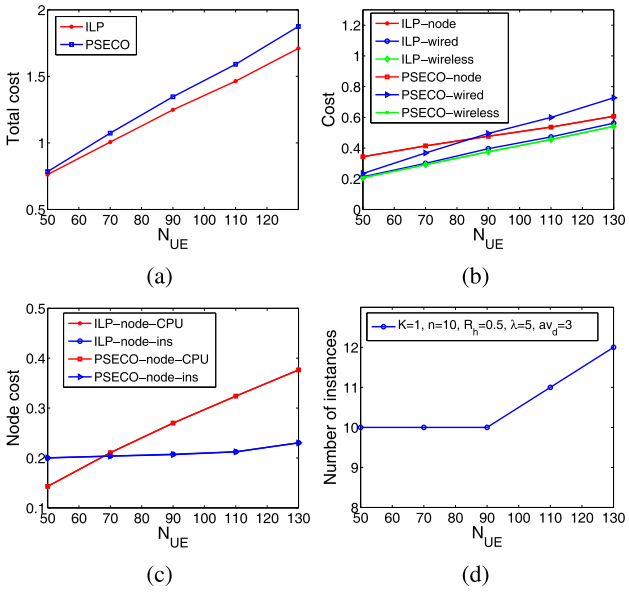

FIGURE 13. Impact of N_{UE} to (a) total cost, (b) different cost, (c) node cost and (d) the number of instances.

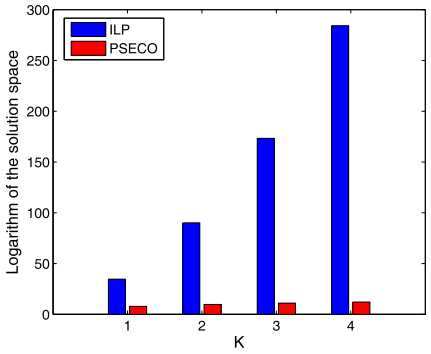

FIGURE 14. Complexity comparison of the two algorithms($n = 10$).

instances, which causes the increase of instantiation cost as shown in Fig. 13(d).

C. COMPLEXITY ANALYSIS

As can be seen from the above simulation results, in terms of the total cost, the ILP based optimization algorithm usually outperforms the heuristic algorithm PSECO. However, from the perspective of complexity, the complexity of the ILP based optimization algorithm is $O(2^{N_{ins}*N_{SBS}})$, while the complexity of PSECO is $O((N_{ins} * N_{SBS})^2)$, where N_{ins} is the number of instances for online game and N_{SBS} is the number of SBSs. The logarithm of the solution space is shown in Fig.14 with the change of K, assuming that there are ten component VNFs for the online game, and each component VNF has one instance, i.e. $N_{ins} = n = 10$. It reveals that the complexity of PSECO grows very slowly, while the complexity of ILP based optimization algorithm grows rapidly. Obviously, the heuristic algorithm PSECO can be used to obtain a near-optimal solution in the large-scale scenario,

especially when the number of application components and the number of the SBSs are large.

VI. CONCLUSION

In this paper, cost minimization of computation offloading for online game is investigated based on probabilistic SFC in fog-enabled heterogeneous radio access network. The problem is formulated as a general ILP problem with the constraints of application maximum tolerable latency, resource limitation and user behavior. In order to reduce the complexity of the ILP based optimization algorithm, a heuristic algorithm called PSECO is proposed. The impact of various parameters to the cost of the two algorithms is evaluated. Simulation results show that PSECO has optimal results with low complexity and it is suitable for large scale networks.

Regarding the future work, on the one hand, mobility of mobile devices and dynamic resource allocation problem are challenging. On the other hand, formulation of air interface plays an important role to the optimization problem, for example, the deployment architecture of fog-enabled radio access network and interference management. In order to improve the efficiency of computation offloading and reduce the cost due to wireless networking, fog-enabled radio access network can be modelled with novel networking techniques such as nested deployed cooperative base stations [39] to improve the capacity of RAN for computation offloading. Furthermore, new interference management techniques like partial interference alignment [40] would be taken into consideration in the future research work.

REFERENCES

[1] F. Chi, X. Wang, W. Cai, and V. C. M. Leung, "Ad-hoc cloudlet based cooperative cloud gaming," *IEEE Trans. Cloud Comput.*, vol. 6, no. 3, pp. 625–639, Jul./Sep. 2018.

[2] S. Choy, B. Wong, G. Simon, and C. Rosenberg, "A hybrid edge-cloud architecture for reducing on-demand gaming latency," *Multimedia Syst.*, vol. 20, no. 5, pp. 503–519, Oct. 2014.

[3] W. Cai, V. C. M. Leung, and L. Hu, "A cloudlet-assisted multiplayer cloud gaming system," *Mobile Netw. Appl.*, vol. 19, no. 2, pp. 144–152, Apr. 2014.

[4] W. Cai *et al.*, "A survey on cloud gaming: Future of computer games," *IEEE Access*, vol. 4, pp. 7605–7620, Aug. 2016.

[5] Y. Lin and H. Shen, "Cloud fog: Towards high quality of experience in cloud gaming," in *Proc.44th Int. Conf. Parallel Process.*, Beijing, China, Sep. 2015, pp. 500–509.

[6] S. Choy, B. Wong, G. Simon, and C. Rosenberg, "EdgeCloud: A new hybrid platform for on-demand gaming," Univ. Waterloo, Waterloo, ON, USA, Tech. Rep. CS-2012-19, 2012.

[7] G. Premsankar, M. Di Francesco, and T. Taleb, "Edge computing for the Internet of Things: A case study," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1275–1284, Apr. 2018.

[8] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1657–1681, 3rd Quart., 2017.

[9] OpenFog Consortium Architecture Working Group, "OpenFog reference architecture for fog computing," Openfog Consortium, Fremont, CA, USA, Tech. Rep. OPFRA001.020817, Feb. 2017.

[10] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A comprehensive survey on fog computing: State-of-the-art and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 416–464, 1st Quart., 2018.

[11] Y. Bi, G. Han, C. Lin, Q. Deng, L. Guo, and F. Li, "Mobility support for fog computing: An SDN approach," *IEEE Commun. Mag.*, vol. 56, no. 5, pp. 53–59, May 2018.

[12] W. Yu *et al.*, "A survey on the edge computing for the Internet of Things," *IEEE Access*, vol. 6, pp. 6900–6919, Nov. 2017.

[13] D. Zhao, D. Liao, G. Sun, and S. Xu, "Towards resource-efficient service function chain deployment in cloud-fog computing," *IEEE Access*, vol. 6, pp. 66754–66766, Oct. 2018.

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

[15] C. Mouradian, S. Kianpisheh, and R. H. Glitho, "Application Component Placement in NFV-based Hybrid Cloud/Fog Systems," in *Proc. IEEE Int. Symp. Local Metrop. Area Netw. (LANMAN)*, Washington, DC, USA, Jun. 2018, pp. 25–30.

[16] T. Wen, H. Yu, and X. Du, "Performance guarantee aware orchestration for service function chains with elastic demands," in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, Berlin, Germany, Nov. 2017, pp. 1–4.

[17] Z. Shaoping, G. Xiujiao, and Y. Hongfang, "Virtual network function instantiation and service function chaining mapping in wide area network," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Chengdu, China, Jul. 2016, pp. 1–6.

[18] R. Riggio, A. Bradai, D. Harutyunyan, T. Rasheed, and T. Ahmed, "Scheduling wireless virtual networks functions," *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 2, pp. 240–252, Jun. 2016.

[19] G. Li, H. Zhou, B. Feng, and G. Li, "Context-aware service function chaining and its cost-effective orchestration in multi-domain networks," *IEEE Access*, vol. 6, pp. 34976–34991, Jun. 2018.

[20] Y. Chen, J. Liu, and Y. Cui, "Inter-player delay optimization in multiplayer cloud gaming," in *Proc. IEEE 9th Int. Conf. Cloud Comput. (CLOUD)*, San Francisco, CA, USA, Jun./ Jul. 2016, pp. 702–709.

[21] S. Zadtootaghaj, S. Schmidt, and S. Möller, "Modeling gaming QoE: Towards the impact of frame rate and bit rate on cloud gaming," in *Proc. 10th Int. Conf. Qual. Multimedia Exper. (QoMEX)*, Cagliari, Italy, May/ Jun. 2018, pp. 1–6.

[22] H.-J. Hong, D.-Y. Chen, C.-Y. Huang, K.-T. Chen, and C.-H. Hsu, "Placing virtual machines to optimize cloud gaming experience," *IEEE Trans. Cloud Comput.*, vol. 3, no. 1, pp. 42–53, Jan./Mar. 2015.

[23] Y. Deng, Y. Li, R. Seet, X. Tang, and W. Cai, "The server allocation problem for session-based multiplayer cloud gaming," *IEEE Trans. Multimedia*, vol. 20, no. 5, pp. 1233–1245, May 2018.

[24] Y. Zhang, P. Qu, J. Cihang, and W. Zheng, "A cloud gaming system based on user-level virtualization and its resource scheduling," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 5, pp. 1239–1252, May 2016.

[25] W. Cai, C. Zhou, V. C. M. Leung, and M. Chen, "A cognitive platform for mobile cloud gaming," in *Proc. IEEE 5th Int. Conf. Cloud Comput. Technol. Sci.*, vol. 1, Bristol, U.K., Dec. 2013, pp. 72–79.

[26] J. Xu and B. W. Wah, "Concealing network delays in delay-sensitive online interactive games based on just-noticeable differences," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, San Jose, CA, USA, Jul. 2013, pp. 1–6.

[27] W. Cai and V. C. M. Leung, "Decomposed cloud games: Design principles and challenges," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Chengdu, China, Jul. 2014, pp. 1–4.

[28] P. Yu, X. Ma, J. Cao, and J. Lu, "Application mobility in pervasive computing: A survey," *Pervas. Mobile Comput.*, vol. 9, no. 1, pp. 2–17, Feb. 2013.

[29] H. Jin, S. Yan, C. Zhao, and D. Liang, "PMC^{2}O: Mobile cloudlet networking and performance analysis based on computation offloading," *Ad Hoc Netw.*, vol. 58, pp. 86–98, Apr. 2017.

[30] S. Bohez, T. Verbelen, P. Simoens, and B. Dhoedt, "Discrete-event simulation for efficient and stable resource allocation in collaborative mobile cloudlets," *Simul. Model. Pract. Theory*, vol. 50, pp. 109–129, Jan. 2015.

[31] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1628–1656, 3rd Quart., 2017.

[32] *Network Functions Virtualisation (NFV); Architectural Framework*, document ETSI GS NFV 002 V1.2.1 (2014-12), ETSI, 2014.

[33] R. Balakrishnan and I. Akyildiz, "Local anchor schemes for seamless and low-cost handover in coordinated small cells," *IEEE Trans. Mobile Comput.*, vol. 15, no. 5, pp. 1182–1196, May 2016.

[34] *Mobile Edge Computing; Market Acceleration; MEC Metrics Best Practice and Guidelines*, document ETSI GS MEC-IEG 006 V1.1.1 (2017-01), ETSI, 2017.

[35] Gurobi. (Apr. 2019). *Gurobi Optimizer*. [Online]. Available: http://www.gurobi.com
[36] A. Marotta and A. Kassler, "A power efficient and robust virtual network functions placement problem," in *Proc. 28th Int. Teletraffic Congr. (ITC)*, vol. 1, Würzburg, Germany, Sep. 2016, pp. 331–339.
[37] H. Zhang, J. Guo, L. Yang, X. Li, and H. Ji, "Computation offloading considering fronthaul and backhaul in small-cell networks integrated with MEC," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Atlanta, GA, USA, May 2017, pp. 115–120.
[38] S. Yu, X. Wang, and R. Langar, "Computation offloading for mobile edge computing: A deep learning approach," in *Proc. IEEE 28th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Montreal, QC, Canada, Oct. 2017, pp. 1–6.
[39] Q. Wu and Q. Liang, "Increasing the capacity of cellular network with nested deployed cooperative base stations," *IEEE Access*, vol. 6, pp. 35568–35577, Jun. 2018.
[40] L. Wang and Q. Liang, "Partial interference alignment for heterogeneous cellular networks," *IEEE Access*, vol. 6, pp. 22592–22601, Apr. 2018.

HAO JIN received the Ph.D. degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 1996, where she is currently an Associate Professor. Her research interests include future network architecture, optimization of mobile wireless communication, mobile cloud computing, and data mining.

XIAOYING ZHU received the B.Eng. degree from Jilin University, in 2017. She is currently pursuing the master's degree with the Key Laboratory of Universal Wireless Communications, Ministry of Education, Beijing University of Posts and Telecommunications, Beijing, China. Her current research interests include computation offloading in wireless networks, mobile cloud computing systems, and NFV.

CHENGLIN ZHAO received the bachelor's degree in radio technology from Tianjin University, in 1986, the master's degree in circuits and systems, and the Ph.D. degree in communication and information system from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 1993 and 1997, respectively, where he is currently a Professor. His current research interests include emerging technologies of short-range wireless communication, cognitive radios, mobile edge computing, and the Internet of Things.