IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received July 20, 2019, accepted August 18, 2019, date of publication August 29, 2019, date of current version September 11, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2938186

Energy-Latency Aware Offloading for Hierarchical Mobile Edge Computing

BINWEI WU[1], JIE ZENG[2], (Senior Member, IEEE), LU GE[2], XIN SU[2], (Senior Member, IEEE), AND YOUXI TANG[1]

[1]National Key Laboratory of Science and Technology on Communications, University of Electronic Science and Technology of China, Chengdu 611731, China
[2]Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing 100084, China

Corresponding author: Jie Zeng (zengjie@tsinghua.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61771107, in part by the National Key Research and Development Program of China under Grant 2018YFB1801903, and in part by the National Science and Technology Major Project under Grant 2018ZX03001004-003.

ABSTRACT Mobile edge computing (MEC) enhances the computing capacity of resources-poor user equipment (UE) by computational offloading. However, edge clouds suffer from a limited computation capacity, and thus cannot cater for a large amount of offloading requests in periods of high load. To tackle this issue, the hierarchical MEC network is proposed and can utilize the vast resources in the backhaul and backbone networks. Previous studies describe the network layout with a three-tier tree which is not suitable for the realistic implementation. Meanwhile, the influences brought by network congestion on backhaul and backbone links are omitted. Thus, we generalize the assumption on the network layout and propose topology-independent offloading algorithms which can balance the workload over the entire region of the MEC network. In order to relieve the congestion on the backhaul and backbone networks, the task routing is incorporated into the offloading optimization, along with the offloading decision, the transmission power control, and the cloud selection. In order to jointly conduct the offloading optimization, we convert the offloading problem into a multi-source single-destination routing. A distributed offloading approach (i.e., BROA) is developed based on the game theory, in which UE collaborates with each other to minimize the network cost in terms of energy consumption and latency. We theoretically analyze the efficiency of UE collaboration and prove that BROA can achieve the globally optimal solution. Furthermore, an approximate offloading algorithm (i.e., FCOA) is developed which can give a quick solution to adapt to time-varying environments. We theoretically demonstrate the convergence, the accuracy, and the time complexity of FCOA. Numerical results show that the proposed algorithms are superior to conventional offloading schemes.

INDEX TERMS Computation offloading, game theory, generalized network layout, hierarchical mobile edge computing (MEC) network.

I. INTRODUCTION

In the past few years, a large number of resource-hungry mobile applications have emerged, such as gaming, virtual reality, and augmented reality. Current mobile terminals, like smart phones or tablets, endure low capacity on the storage and computation. Deploying resource-hungry applications on mobile terminals results in rapid battery depletion. To tackle this issue, the mobile edge computing (MEC) technique is proposed and enables user equipment (UE) to offload intensive mobile computation tasks to nearby clouds. By leveraging the computational resources on the clouds, the battery lifetime increases and the computation performance of UE is improved. However, the task offloading results in extra overhead, e.g., additional energy consumption and latency for the uplink transmission. Thus, advanced offloading schemes are required to improve the efficiency of the MEC network.

The conventional MEC network assumes that clouds are deployed at the edge of the network [1]. Plenty of researches have been done on offloading optimization, which improves network performance in terms of energy consumption and latency [2]–[4]. However, edge clouds suffer from limited computation capacity, and thus cannot cater for a large

The associate editor coordinating the review of this article and approving it for publication was Zhibo Wang.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see http://creativecommons.org/licenses/by/4.0/

amount of offloading requests in periods of high load [5]. Recent studies have proposed a hierarchical MEC architecture to tackle this issue [6]. Clouds are deployed in multiple tiers which correspond to different network layers (i.e., access layer, aggregate layer, and core layer). Higher tiers consist of more powerful clouds, and can receive migration tasks from lower tiers in case of overload. Vast resources in the backhaul and backbone networks (e.g., data center) can be leveraged. Formal analysis and results demonstrate the superiority of the hierarchical MEC network in terms of latency and energy consumption [7], [8].

Most studies on the hierarchical MEC network use a three-tier tree to describe the network layout [6]–[9]. The tree topology can reflect the hierarchy of the mobile network. However, in real-world implementations, the topology of the mobile network is more complex since the tree structure will suffer from some disadvantages, e.g., vulnerability, non-resilient [10], [11]. Extensional researches have been done on the topology design and planning, in which the tree model is usually combined with other structures (e.g., rings) [11]. Practical instances of the mobile network can be found in [12] which also implies that pure tree structures are not suitable for realistic implementations. Therefore, a generalized network topology needs to be considered.

Since the network topologies are generalized, we need to propose topology-independent offloading algorithms. Previous studies such as [7], [8] propose offloading algorithms which improve the offloading performance by migrating tasks from lower tiers to higher tiers. Tasks are not balanced inside the tier due to the tree structure (clouds in the same tier are not directly connected with each other). We notice that it is difficult to define an explicit association between clouds and tires in generalized network topologies. Thus, the algorithms in [7], [8] cannot be directly migrated to the cases with generalized network topologies. The task migrations (or cloud selection which determines the target clouds of UE) need to be conducted over the entire region of the network.

Meanwhile, the impacts brought by the resource-constraint links and network devices in the backhaul and backbone networks have to be taken into consideration. Previous studies assume that the transmission latency on the backhaul and backbone networks are constant regarding the workload [5], [8], [13]. However, resource-constraint links and network devices (e.g., switches, routers) are easy to be overloaded in periods of high load. The congestion will damage the network performance with an increase of transmission delay and a slowdown of throughput. In hierarchical MEC networks, the damages would become more significant since the tasks may experience a long propagation distance [1]. We address this issue by designing a task routing strategy which allows the task flows to bypass the congestion links and devices.

It is noticed that the generalization of the network topology and the incorporation of the task routing will significantly increase the dimensionality of the offloading problem. To propose a practical offloading algorithm, several challenges need to be addressed. Firstly, the proposed offloading algorithm needs to work efficiently even in a large-scale network. Centralized offloading algorithms may not be suitable because of some drawbacks, e.g., a single point of failure, overload on the central coordinator. Secondly, the time-varying UE demands and wireless channel coefficients require that the proposed algorithm is able to produce a quick and acceptable solution even with a high-dimensional solution space. Thirdly, a theoretical performance analysis is suggested to be given. The theoretical analysis not only can give a credible assessment on algorithm performance but also can be used as a basis for the parameters tuning.

To address the above-mentioned issues, we carry out the following work. We utilize a direct graph to describe the layout of the hierarchical MEC network so that the proposed algorithms are topology-independent. The workload is balanced over the entire region of the mobile network through cloud selection. We incorporate the task routing into the offloading optimization to emphasize the impacts brought by the congestion in the backhaul and backbone networks. Distributed offloading algorithms are proposed based on the game theory, in which no central coordinators are involved. To give a quick solution in time-vary environments, we propose a fast-converged algorithm by introducing an approximate factor. We conduct theoretical analysis on offloading performance with auxiliary functions which can track global influences brought by individual UE behaviors. The main contributions in this paper are listed as follows:

- We formulate the offloading problem in the hierarchical MEC network to minimize network cost in terms of energy consumption and latency. A generalized network topology model described by a direct graph is considered in the proposed offloading problem.
- We jointly consider the optimization on the offloading decision, the uplink transmission power control, the cloud selection, and the task routing. To carry out the optimization in a joint manner, we convert the optimization into a multi-source single-destination routing problem.
- We propose a distrusted energy-latency aware offloading algorithm (BROA) based on the game theory. A marginal payoff function is defined so that the collaboration among UE can lead to global improvement on offloading performance. We show that BROA can achieve a globally optimal solution.
- In order to make the algorithm suitable for the realistic environment, we propose an approximate algorithm (FCOA) which accelerates the convergence speed. The convergence, accuracy, and time complexity of FCOA are theoretically analyzed.

The rest of this paper is organized as follows. We review the related work on the offloading and introduce the game theory in Section II. We describe the system model and formulate the offloading problem in Section III. In Section IV, we analyze the problem and solve it with a game theory-based approach. In Section V, we propose an approximate algorithm FCOA and give a theoretical analysis of its performance.

Simulation results are shown in Section VI. Finally, we conclude the paper in Section VII.

II. RELATE WORK AND BACKGROUND

A. COMPUTATION OFFLOADING

It is generally accepted that MEC will play an important role in various 5G applications, such as video stream analysis service [14], augmented reality service [15], [16], IoT applications [17], connected vehicles [18]. Due to its significant impacts on the mobile wireless network, MEC obtains a lot of attention from operators and vendors. European 5G Infrastructure Public Private Partnership (5GPPP) has recognized the MEC as one of the key emerging technologies for 5G networks [19], [20]. In April 2017, 3GPP has included supporting edge computing as one of the high-level features in 5G systems [21].

Initial studies on MEC assume that the clouds are deployed at the edge of the mobile network, co-located with APs. In order to minimize energy consumption and latency, the authors of [22] propose an offloading strategy in a single-user MEC environment. Similar work has been done in [23], [24], which take the computation-deadline constraints into consideration. The case with multiple servers is considered in [25], in which the authors formulate the offloading optimization as a multiple knapsack problem. Other than the offloading decision, the uplink transmission power control is considered to further improve the offloading performance [26]. The authors of [27] jointly consider the offloading decision and physical resource block (PRB) allocation to minimize the energy consumption of UE in a multi-user single-server MEC model. Cases with multiple clouds in heterogeneous networks are considered in [28]. The authors of [29] study the effects brought by booting and provisioning time of servers. Overheads incurred by collaboration communication are investigated in [30]. The offloading gain is further exploited by combining with additional advanced techniques (e.g., energy harvest [3], wireless power transfer [4], unmanned aerial vehicle [31]).

The authors of [32] point out that the concept of clouds in the MEC network is expected to be supported by a 3-tier network. The vast resources in the backhaul and backbone networks (e.g., data center) can be leveraged to improve the offloading gain. The preliminary studies are conducted in [33]. They introduce a cloud network planning approach which optimally places the cloud facilities among a given set of available sites and assigns a set of APs to the clouds.

Motivated by [33], the authors of [6] firstly propose a hierarchical MEC architecture in accordance with the principles of the LTE-advanced network. Clouds are classified into the edge clouds and the central clouds based on their proximity to mobile devices. The authors develop an auction-based algorithm for resource allocation to maximize the profit of providers from the economic perspective. Similar work has been done in [13] which supports the coexistence of centralized clouds and edge clouds by integrating the FiWi networks. Meanwhile, a code partition scheme is proposed in [7] to minimize the network cost in terms of energy consumption and overall latency. In order to reduce the average response time, the authors of [8] develop a workload allocation scheme which determines the UE-Cloud assignment and the resource provision. Computational resource allocation on the clouds are considered in [5] to minimize the operator's cost and UE's energy consumption. It is noticed that the underlying networks in these studies are described by the tree structure, which omits the complicacy of the realistic environment.

Other than the optimization on offloading performance, security is another big issue. The MEC network is essentially a distributed system which is vulnerable to various attacks. The attacks can be divided into jamming, DoS, spoofing attacks, man-in-the-middle attacks, and privacy leakage [34]. To address these security threats, both UE and operators have to make a number of decisions [35]. UE observes the received jamming signaling and chooses the offloading policies (e.g., cloud selection, offloading rate) accordingly. Some researchers concentrate on the network side. In their work, the clouds are responsible for the fast detection of spoofing messages and rogue users [36]. PHY-authentication technique is used in [37] to provide lightweight protection against identity-based attacks without leaking UE privacy.

B. ALGORITHM DESIGN

Most studies formulate the MEC offloading problem as mixed-integer linear programming (MILP) since they assume the latency and energy consumption on a particular network element are constant [38]. Mixed-integer non-linear programming (MINLP) model is also used when the authors use precise models [2], [8], [9]. Since the offloading optimization is often integrated with other techniques, the solution space of the final problem is high-dimensional. Therefore, it is difficult to obtain a quick and acceptable answer.

Some researchers use the exhaust algorithm (e.g., enumeration method [39]) or ILP solver (e.g., branch and bound [5], CPLEX, etc.) to obtain a precise solution. These algorithms are highly complex and obviously cannot be directly used in realistic scenarios. Thus, most researchers use the heuristic-based algorithms (e.g., heuristic search [2], [13], genetic algorithm, particle swarm optimization, etc.) which reduce the complexity [31]. The advantage of these algorithms is that they can offer a quick solution. However, it is hard to conduct a theoretical analysis for the heuristic algorithms. Offloading performance cannot be guaranteed. To relieve this issue, the authors of [40] develop a hybrid algorithm. The scheduler identifies a set of conditions and builds an algorithm that performs almost optimally for each condition. However, the identification of conditions is highly empirical, which significantly affects the availability of the algorithm. Sophisticated algorithms, such as reinforcement learning, are also used in the field of MEC offloading [41]. These algorithms traverse and analyze the problem space to find a better solution. However, it is still questionable whether the current learning algorithm is able to deliver a quick and acceptable solution in a distributed large-scale problem.

Sometimes, heuristic-based algorithms are integrated with the decomposing technique, such as [2], [8], [9], [13], [42]. These studies decompose the offloading problem into subproblems, such as computation resource allocation, communication resource allocation, offloading decision, and cloud selection. Then, the authors conduct the optimization in an isolated manner which usually results in performance degradation. Additionally, from the perspective of implementations, most of these offloading algorithms need a centralized coordinator which is easy to be overloaded in the large-scale network.

Distributed offloading algorithms have been proposed based on the non-cooperative game which is a special branch of game theory [13], [38], [43]. The non-cooperative game has been widely used to analyze strategic interactions between rational decision-makers. In these game-theory based algorithms, UE coordinates with each other to achieve a common goal (e.g., latency, energy consumption). However, most game theory-based algorithms apply a greedy and selfish revision rule for their players, which leads to unexpected results (e.g., nonconvergence, performance degradation) [44].

C. GAME THEORY

In this paper, we mainly focus on the non-cooperative routing game which occurs in a multi-commodity flow network [44]. Normally, the network is described by a direct graph $\overline{\mathcal{G}} = \{\overline{\mathcal{V}}, \overline{\mathcal{E}}\}$ where $\overline{\mathcal{V}}$ and $\overline{\mathcal{E}}$ are the vertices and edges, respectively. Each player u with a demand r_u is associated with a source-sink pair $(v_u^{\text{source}}, v_u^{\text{dest}})$. We call such pairs as commodities, p.s., each player is identified with one commodity. It is noticed that different players can originate from different source vertices and travel to different sink vertices. We use $\mathcal{F}_u$ to denote the feasible v_u^{source}–v_u^{dest} paths of the network and define $\mathcal{F} = \cup \mathcal{F}_u$. We allow $\overline{\mathcal{G}}$ to contain parallel paths and a vertex can participate in multiple source-sink pairs. Each vertex and edge incurs a routing cost which increases with their congestion. The cost of the path is defined as the sum of the routing cost of the constituent vertices and edges. In the routing game, each player attempts to minimize its routing cost.

There are two types of routing game, i.e., nonatomic routing and atomic routing [44], [45]. In the nonatomic routing, each commodity represents a large population of individuals, each of whom controls a negligible amount of traffic. In the atomic routing, each commodity represents a single player who must route a significant amount of traffic on a single path. The nonatomic routing is used in Section IV to analyze the offloading performance of BROA. Players change the amount of traffic on separate paths to improve their utilities. In Section V, we use atomic routing game to analyze the influences brought by the approximate factor.

The equilibrium is a proposed solution of a game [44]. At the point of the equilibrium, no player has something to gain by changing only their own strategy. The major concerns in the field of the routing game are the existence and efficiency of the equilibrium. To tackle this issue, we use the tools in the field of the potential game [46]. A game is said to be a potential game if the incentive of all players to change their strategy can be expressed using a single global function called the potential function. Since the incentives of all players are mapped into one function, the set of equilibrium can be found by locating the local optima of the potential function. The convergence property of an iterated game and the efficiency of the equilibrium can be analyzed by studying the potential function.

III. SYSTEM MODEL AND PROBLEM FORMATION

A. NETWORK MODEL

We adopt the network model illustrated in Figure 1. It consists of UE (i.e., $\mathcal{U}$), APs (i.e., $\mathcal{B}$), aggregate nodes, core nodes, and MEC servers (or clouds). In the hierarchical MEC network, the clouds at the edge of the network are called edge clouds while the others are central clouds. UE can either offload their computation tasks to the remote clouds or process them locally. The notions about the system model in this section are listed in Table 1.

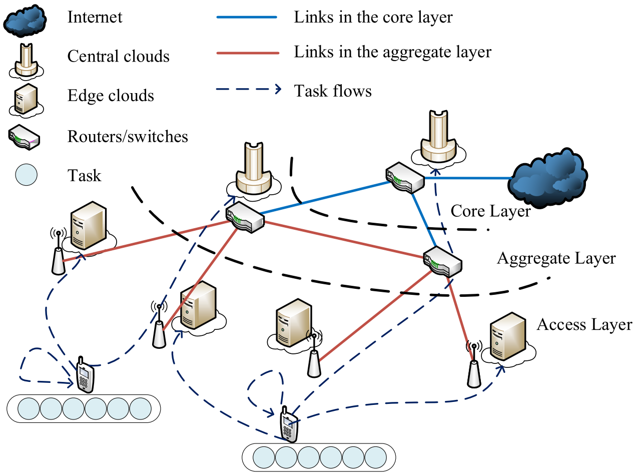


FIGURE 1. The MEC network with heterogeneous clouds.

We consider a generalized network topology, which is described by a direct graph, $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$. $\mathcal{V}$ and $\mathcal{E}$ are the sets of network elements (NEs) and links, respectively. We map the physical appliances to different NEs based on their functions. NEs consist of the processing units (i.e., $\mathcal{V}^{\text{mec}}$) and forwarding units (i.e., $\mathcal{V}^{\text{rou}}$). The processing units provide MEC services and the forwarding units perform the traffic directing functions. $\mathcal{E}$ consists of wired links which connect the NEs.

UE accesses the network with the orthogonal multiple access (OMA) technique. Each UE, $u \in \mathcal{U}$, is equipped with a single antenna with multiple transmission power levels. The transmission power level is denoted as $\mathcal{M} = \{1, \cdots, M\}$, where $m \in \mathcal{M}$ corresponds to a fixed transmission power $p_{u,m}$. The capacity of wireless channels from u to

TABLE 1. Notions of the system model.

Symbols	Descriptions
$\mathcal{U}, \mathcal{B}$	Sets of UE and APs
$\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$	Network topology consists of NEs and links
$\mathcal{V}^{\text{rou}}$, $\mathcal{V}^{\text{mec}}$	Set of forwarding units and processing units
$f \in \mathcal{F}$	Path from UE to processing units
K_j^{mec}, K_e^{link}, $K_{u,0}$	Capacity of processing units j, links e, and local computing on u
$K_{u,m}$	Channel capacity under power $p_{u,m}$
α	Fraction of idle energy consumption
$E_j^{\max}$	Energy consumed of a fully-utilized server
$\xi_{j,1}$, $\xi_{j,2}$	Parameters for forwarding units in its latency model
$\xi_{j,3}$, $\xi_{j,4}$	Parameters for processing units in its latency model
ς_u	A constant related to UE architecture
w_j, w_e	Aggregate traffic loads on NEs and links
ω_u, L, κ	Task rate of u, task length, required CPU cycles per bits
D	Maximum hop for the task routing
Γ	A discounting for wireless channel capacity
$h_{u,b}$, σ^2, B_u^{oma}	Channel coefficient, power of noise, wireless transmission bandwidth
$\mathcal{M}$, $p_{u,m}$	Set of feasible transmission power, m^{th} transmission power of u
γ_d	Coefficient reflects the importance among energy and latency
γ_e	Coefficient reflects the importance of energy on users and network
$x_u = \{x_{u,m}\}_{m \in \{0\} \cup \mathcal{M}}$	Decision parameters for offloading of u
$X = \{x_u\}_{u \in \mathcal{U}}$	Decision parameters for offloading
$y_u = \{y_{u,f}\}_{f \in \mathcal{F}}$	Decision parameters for routing of u
$Y_u = \{y_u\}_{u \in \mathcal{U}}$	Decision parameters for routing

$b \in \mathcal{B}$ under $p_{u,m}$ is represented as

$$K_{u,m} = B_u^{\text{oma}} \log_2(1 + \frac{p_{u,m}\left|h_{u,b}\right|^2}{\Gamma \cdot \sigma^2}), \tag{1}$$

where $h_{u,b}$ and σ^2 are the coefficient of the fading channel and the power of white noise, respectively. B_u^{oma} is the spectrum bandwidth and $\Gamma \geq 1$ is a constant accounting for the gap from the channel capacity due to a practical coding and modulation scheme.

B. USERS AND TASK FLOWS

A UE, $u \in \mathcal{U}$, generates the computational tasks that arrive at the rate of ω_u. The data size of a task is L. Each task has a required CPU cycle of κ per bit. UE can either offload their tasks to the processing units within their converge (D, measured by hops) or process it locally.

We consider the data-partition model for the offloading, in which the tasks are bit-wise independent and can be arbitrarily divided into different groups and executed by different entities in MEC systems [1]. We use $\boldsymbol{x}_u = \{x_{u,m}\}_{m \in \{0\} \cup \mathcal{M}}$ to represent the offloading decision, where $x_{u,m}$ is the ratio of the tasks offloaded with power $p_{u.m}$. Accordingly, we have $\sum_{m=0}^{|\mathcal{M}|} x_{u,m} = 1$. The offloading decision of the entire network is denoted as $\boldsymbol{X} = \{\boldsymbol{x}_u\}_{u \in \mathcal{U}}$.

We use $\mathcal{F}$ to denote the feasible paths from UE to processing units. A path, $f \in \mathcal{F}$, consists of a sequence of forwarding units (e.g., APs, routers), a series of links and a processing unit. We use $\boldsymbol{Y} = \{\boldsymbol{y}_u\}_{u \in \mathcal{U}}$ to denote a flow, where $\boldsymbol{y}_u = \{y_{u,f}\}_{f \in \mathcal{F}}$ is nonnegative vector indexed by $\mathcal{F}$. $y_{u,f}$ is the ratio of tasks which come from u and choose the path f. Given a flow $\boldsymbol{Y}$, the aggregate loads on NEs (i.e., w_j) and links (i.e., w_e) are represented as

$$w_j = \sum_{u \in \mathcal{U}} \sum_{\{f : j \in f\}} y_{u,f} \omega_u L, \quad \forall j \in \mathcal{V}, \tag{2}$$

and

$$w_e = \sum_{u \in \mathcal{U}} \sum_{\{f : e \in f\}} y_{u,f} \omega_u L, \quad \forall e \in \mathcal{E}. \tag{3}$$

C. ENERGY CONSUMPTION AND AVERAGE RESPONSE TIME

The models in this section are mainly adopted from [1]. We assume that the energy is mainly consumed by the wireless transmission and the task processing.

The energy consumption on $j \in \mathcal{V}^{\text{mec}}$ is represented as

$$E_j\left(\boldsymbol{X}, \boldsymbol{Y}\right) = \alpha E_j^{\max} + (1 - \alpha) \frac{E_j^{\max}}{K_j^{\text{mec}}} w_j \kappa, \tag{4}$$

where $E_j^{\max}$ is the energy consumption of a fully-utilized server and α is the fraction of the idle energy consumption. K_j^{mec} represents the maximum computation capacity of j, measured by CPU cycles per time slot.

The energy consumption for local computing is denoted as

$$E_u^{\text{comp}}\left(\boldsymbol{X}\right) = x_{u,0} \omega_u \cdot \varsigma_u \kappa L \cdot K_{u,0}^2, \quad \forall u \in \mathcal{U}, \tag{5}$$

where $K_{u,0}$ is the maximum capacity of local computing on u. ς_u is a constant related to u's architecture.

The energy consumption motivated by the wireless transmission is represented as

$$E_u^{\text{trans}}\left(\boldsymbol{X}\right) = \sum_{m \in \mathcal{M}} \left(x_{u,m} \omega_u \frac{p_{u,m} L}{K_{u,m}} \right), \quad \forall u \in \mathcal{U}. \tag{6}$$

The task response time consists of the transmission delay and processing delay. The transmission delay is incurred by $\mathcal{V}^{\text{rou}}$, $\mathcal{E}$ and wireless channels. The processing delay is motivated by $\mathcal{V}^{\text{mec}}$ and UE's local computing.

We represent the overall latency on the wired link $e \in \mathcal{E}$ as $D_e^{\text{comm}}\left(\boldsymbol{X}, \boldsymbol{Y}\right) = w_e / K_e^{\text{link}}$, where K_e^{link} stands for the capacity of e.

For the transmission latency on $\mathcal{V}^{\text{rou}}$, we adopt a linear latency model since buffer utilization and packet loss increases as the bitrate grows [47]. The overall transmission latency on $j \in \mathcal{V}^{\text{rou}}$ is expressed as

$$D_j^{\text{comm}}(\boldsymbol{X}, \boldsymbol{Y}) = \left(\xi_{j,1} w_j + \xi_{j,2}\right) w_j, \tag{7}$$

where $\xi_{j,1}$ and $\xi_{j,2}$ are parameters for the linear model, depending on the characteristic of j.

We represent the overall transmission latency on wireless channels as $D_{u,m}^{\text{comm}}(\boldsymbol{X}) = x_{u,m}\omega_u \cdot L/K_{u,m}$.

We assume that the resources on the processing units are managed in the form of VMs. Thus, we represent the overall latency on $j \in \mathcal{V}^{\text{mec}}$ in (8) [48].

$$D_j^{\text{comp}}(\boldsymbol{X}, \boldsymbol{Y}) = \frac{\xi_{j,3}}{K_j^{\text{mec}}}\left(1 + \frac{\xi_{j,4} w_j}{K_j^{\text{mec}}}\right) w_j, \tag{8}$$

where $\xi_{j,3}$ and $\xi_{j,4}$ are constants associated with j's structure, e.g, the number of virtual machines (VM).

The overall processing latency on UE is given by

$$D_u^{\text{comp}}(\boldsymbol{X}) = x_{u,0}\omega_u \frac{\kappa L}{K_{u,0}}, \quad \forall u \in \mathcal{U}. \tag{9}$$

D. PROBLEM FORMATION

In this paper, we focus on the minimization of the network cost incurred by energy consumption and overall response time. The energy consumption cost, $E(\boldsymbol{X}, \boldsymbol{Y})$, is represented as

$$\begin{aligned} E(X, Y) = {} & \gamma_e \sum_{j \in \mathcal{V}^{\text{mec}}} \left[\alpha E_j^{\max} + (1-\alpha) E_j^{\max} \frac{\kappa w_j}{K_j^{\text{mec}}}\right] \\ & + (1-\gamma_e) \sum_{u \in \mathcal{U}} x_{u,0}\omega_u \cdot \varsigma_u \kappa L \cdot K_{u,0}^2 \\ & + (1-\gamma_e) \sum_{u \in \mathcal{U}} \sum_{m \in \mathcal{M}} x_{u,m}\omega_u \frac{p_{u,m} L}{K_{u,m}}, \end{aligned} \tag{10}$$

where γ_e reflects the relative importance of energy consumptions between UE and the network. The expected overall response time $D(\boldsymbol{X}, \boldsymbol{Y})$ is represented as

$$\begin{aligned} D(X, Y) = {} & \sum_{e \in \mathcal{E}} \frac{w_e}{K_e^{\text{link}}} + \sum_{j \in \mathcal{V}^{\text{rou}}} w_j \left(\xi_{j,1} w_j + \xi_{j,2}\right) \\ & + \sum_{j \in \mathcal{V}^{\text{mec}}} \frac{\xi_{j,3}}{K_j^{\text{mec}}} \left(1 + \frac{\xi_{j,4} w_j}{K_j^{\text{mec}}}\right) w_j \\ & + \sum_{u \in \mathcal{U}} \left[x_{u,0}\omega_u \frac{\kappa L}{K_{u,0}} + \sum_{m \in \mathcal{M}} \left(x_{u,m}\omega_u \frac{L}{K_{u,m}}\right)\right]. \end{aligned} \tag{11}$$

Then, we define the network cost, $\Phi(\boldsymbol{X}, \boldsymbol{Y})$, in (12).

$$\Phi(\boldsymbol{X}, \boldsymbol{Y}) = \gamma_d E(\boldsymbol{X}, \boldsymbol{Y}) + (1 - \gamma_d) D(\boldsymbol{X}, \boldsymbol{Y}), \tag{12}$$

where γ_d is the coefficient which identifies the relative importance between the overall response time and the energy consumption. It is also noticed that (12) can be considered as a weighted sum approach of a general multi-objective optimization problem, i.e., minimizing $E(\boldsymbol{X}, \boldsymbol{Y})$ and $D(\boldsymbol{X}, \boldsymbol{Y})$.

We consider the computation offloading in hierarchical MEC network from the following aspects: 1) the offloading decision of the tasks; 2) the UE uplink transmission power control; 3) the association and routing between UE and clouds. Thus, we formulate the computation offloading problem in P1.

$$\begin{aligned} \text{P1}: \quad & \min_{\boldsymbol{X}, \boldsymbol{Y}} \quad \Phi(\boldsymbol{X}, \boldsymbol{Y}) && \text{(13a)} \\ \text{s.t.} \quad & w_v \le B_v^{\text{bw}}, \quad \forall v \in \mathcal{V}^{\text{rou}} && \text{(13b)} \\ & w_e \le B_e^{\text{bw}}, \quad \forall e \in \mathcal{E} && \text{(13c)} \\ & \kappa w_j \le K_j^{\text{mec}}, \quad \forall j \in \mathcal{V}^{\text{mec}} && \text{(13d)} \\ & \kappa x_{u,0}\omega_u L \le K_{u,0}, \quad \forall u \in \mathcal{U} && \text{(13e)} \\ & |f| \le D, \quad \forall f \in \mathcal{F} && \text{(13f)} \\ & \sum_{m=0}^{M} x_{u,m} = 1, \quad \forall u \in \mathcal{U} && \text{(13g)} \\ & x_{u,m} \ge 0, \quad \forall u \in \mathcal{U} && \text{(13h)} \\ & \sum_{m \in \mathcal{M}} x_{u,m}\omega_u = \sum_{f \in \mathcal{F}} y_{u,f}, \quad \forall u \in \mathcal{U}. && \text{(13i)} \end{aligned}$$

Constraints (13b) and (13c) guarantee that the aggregated bandwidth consumptions on $\mathcal{V}^{\text{rou}}$ and links are less than its maximum capacity (i.e., $B_v^{\text{bw}}, B_e^{\text{bw}}$ in bps, respectively). Constraint (13d) and (13e) guarantee the workloads on the clouds and the UE would not exceed their maximum capacity. Constraint (13f) gives a maximum hop constraint for the task flows. Constraint (13g), (13h), and (13i) give the mathematical completeness of $\boldsymbol{X}$ and $\boldsymbol{Y}$.

IV. THE GAME THEORY-BASED APPROACH FOR ENERGY-EFFICIENT OFFLOADING

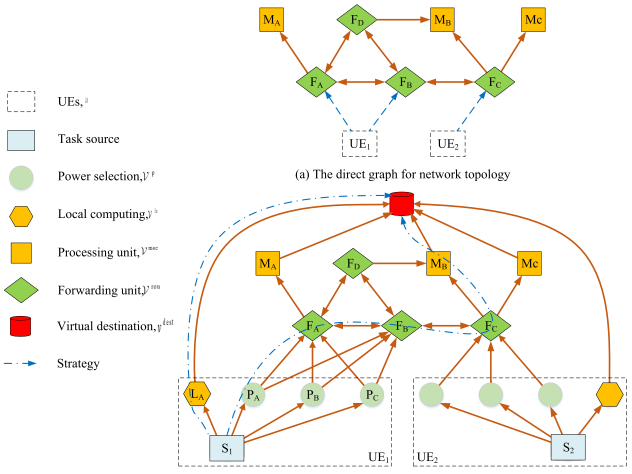
A. ANALYSIS OF THE ENERGY-EFFICIENT OFFLOADING PROBLEM

In this section, we convert P1 into a routing problem so that the offloading decision, the transmission power control, the task routing, and the cloud selection can be jointly optimized. The key notions in the following sections are abstracted in Table 2.

We extend $\mathcal{G}$ into $\overline{\mathcal{G}} = \{\overline{\mathcal{V}}, \overline{\mathcal{E}}\}$. An instance of $\mathcal{G}$ and $\overline{\mathcal{G}}$ are shown in Figure 2(a) and (b), respectively. As shown in Figure 2, we decompose each physical UE (dashed rectangles) into following parts, i.e., a vertex representing the task source (rectangles with baby blue), a vertex representing the local computing ($\mathcal{V}^{\text{lc}}$, marked by hexagons), and vertices representing the different transmission power levels ($\mathcal{V}^{\text{p}}$, marked by circles). A virtual destination (v^{dest}) is introduced so that all commodities can share the same destination. Then, we have $\overline{\mathcal{V}} = \mathcal{V} \cup \mathcal{V}^{\text{lc}} \cup \mathcal{V}^{\text{p}} \cup v^{\text{dest}}$. Four types of edges are added to get a path from the task sources to v^{dest}: 1) edges inside the UE which links the tasks sources to their $\mathcal{V}^{\text{lc}}$ and $\mathcal{V}^{\text{p}}$; 2) edges among $\mathcal{V}^{\text{lc}}$ and v^{dest}, 3) edges among $\mathcal{V}^{\text{mec}}$ and v^{dest}; 4) edges among $\mathcal{V}^{\text{p}}$ and their available APs.

TABLE 2. Notions of the proposed games.

Symbols	Descriptions
$\overline{\mathcal{G}} = \{\overline{\mathcal{V}}, \overline{\mathcal{E}}\}$	Extended graph for the game model
$\mathcal{V}^{\mathrm{p}}, \mathcal{V}^{\mathrm{lc}}, v^{\mathrm{dest}}$	Sets of nodes for power selections, local computing, and virtual destination
$\delta_v^{\mathrm{rou}}, \delta_v^{\mathrm{mec}}, \delta_v^{\mathrm{lc}}, \delta_v^{\mathrm{p}}$	Indicators for the forwarding units, processing units, local computing nodes, and power selections nodes
$\delta_e^{\mathcal{E}}$	Indicators for the wired links
c_v^{sys}, c_v^{τ}, and c_e^{τ}	Energy costs on node v, delay costs on node v, and link e
$G = \{\mathcal{U}, \mathcal{S}, U(s)\}$	The proposed offloading game
r_u	The demands of u
$\mathcal{S}$	Feasible flows
$s_u = \{s_{u,f}\}_{f \in \mathcal{F}}$	The strategy of player u, which is a nonnegative vector indexed by $\mathcal{F}$
$s = \{s_u\}_{u \in \mathcal{U}}$	The aggregate actions of all players
$\overline{s}$	The equilibrium of the proposed game
β	Approximate factor
$G^{\mathrm{vir}} = \{\mathcal{I}, \mathcal{F}, U^{\mathrm{vir}}(s^{\mathrm{vir}})\}$	The proposed approximate offloading game
$\mathcal{I}, \mathcal{I}_u$	Set of virtual tasks, set of virtual tasks associated with u
r_i^{vir}	Demands of i in G^{vir}
s_i^{vir}	The strategy of i, which is a path containing nodes and links
s_{-i}^{vir},s^{vir}	The joint strategies for the players other than i, the aggregate actions of all virtual tasks
$\widetilde{s}_i^{\mathrm{vir}}$	The strategy of i in the next iteration
$\overline{s}^{\mathrm{vir}}$, $\widehat{s}^{\mathrm{vir}}$	The equilibrium in G^{vir}, the optimal solutions regarding the network costs
$\overline{w}_v$, $\overline{w}_e$	The aggregate traffic loads on v and e at $\overline{s}^{\mathrm{vir}}$
$\hat{w}_v$, $\hat{w}_e$	The aggregate traffic loads on v and e at $\hat{s}^{\mathrm{vir}}$


M_A
F_D
M_B
Mc
F_A
F_B
F_C
UE_1
UE_2
UEs, $\mathcal{U}$
Task source
Power selection, $\mathcal{V}^{\mathrm{p}}$
Local computing, $\mathcal{V}^{\mathrm{lc}}$
Processing unit, $\mathcal{V}^{\mathrm{mec}}$
Forwarding unit, $\mathcal{V}^{\mathrm{rou}}$
Virtual destination, v^{dest}
Strategy
L_A
P_A
P_B
P_C
S_1
S_2

(a) The direct graph for network topology

(b) The extended graph for the offloading game

FIGURE 2. Graphs for the energy-latency aware offloading problem.

It is noticed that $\overline{\mathcal{G}}$ is compliant to any network configuration by adding/deleting/modifying vertices and edges.

We define a nonatomic routing procedure in $\overline{\mathcal{G}}$. Each task source tries to route its demand $\omega_u L$ to v^{dest}. The vertices and edges along the path will introduce a routing cost. We use δ_v^{rou}, δ_v^{mec}, δ_v^{lc}, δ_v^{p} as $\{0, 1\}$ indicators for $\mathcal{V}^{\mathrm{rou}}$, $\mathcal{V}^{\mathrm{mec}}$, $\mathcal{V}^{\mathrm{lc}}$, $\mathcal{V}^{\mathrm{p}}$, e.g., $\delta_v^{\mathrm{rou}} = 1, \forall v \in \mathcal{V}^{\mathrm{rou}}$, vice versa. Similarly, we use $\delta_e^{\mathcal{E}}$ to indicate whether e is in $\mathcal{G}$ or not. w_e and w_v are used to denote the aggregate loads on edges and vertices, respectively. The routing cost of an edge e is defined $c_e^{\tau}(w_e) = \gamma_d w_e \delta_e^{\mathcal{E}} / K_e^{\mathrm{link}}$, $e \in \overline{\mathcal{E}}$ and $K_e^{\mathrm{link}} = \infty, \forall e \in \overline{\mathcal{E}} \backslash \mathcal{E}$. The routing cost incurred by vertices consists of two parts, i.e., $c_v^{\mathrm{sys}}(w_v)$ and $c_v^{\tau}(w_v)$, which are defined in (14) and (15), separately.

$$\begin{aligned} c_v^{\mathrm{sys}}(w_v) &= \gamma_d \cdot \gamma_e \left(1 - \alpha\right) E_v^{\max} \frac{\kappa}{K_v} \delta_v^{\mathrm{mec}} \\ &\quad + \gamma_d \left(1 - \gamma_e\right) \varsigma_u \kappa K_{u,0}^2 \delta_v^{\mathrm{lc}} \\ &\quad + \gamma_d \left(1 - \gamma_e\right) \frac{p_{u,m}}{K_{u,m}} \delta_v^{\mathrm{p}}, \quad v \in \overline{\mathcal{V}} \qquad (14) \end{aligned}$$

and

$$\begin{aligned} c_v^{\tau}(w_v) &= \left(1 - \gamma_d\right) \left(\xi_{v,1} w_v + \xi_{v,2}\right) \delta_v^{\mathrm{rou}} \\ &\quad + \left(1 - \gamma_d\right) \frac{\xi_{j,3}}{K_j^{\mathrm{mec}}} \left(1 + \frac{\xi_{j,4} w_j}{K_j^{\mathrm{mec}}}\right) \delta_v^{\mathrm{mec}} \\ &\quad + \left(1 - \gamma_d\right) \frac{\kappa}{K_{u,0}} \delta_v^{\mathrm{lc}} + \frac{\left(1 - \gamma_d\right)}{K_{u,m}} \delta_v^{\mathrm{p}}, \quad v \in \overline{\mathcal{V}} \quad (15) \end{aligned}$$

It is noticed that for link $e \in \mathcal{E}$, $c_e^{\tau}(w_e)$ equals the expected latency when the task flows travel through. For $v \in \mathcal{V}^{\mathrm{mec}} \cup \mathcal{V}^{\mathrm{lp}}$, $c_v^{\mathrm{sys}}(w_v)$ equals the energy consumption and $c_v^{\tau}(w_v)$ equals the processing time. For $v \in \mathcal{V}^{\mathrm{p}}$, $c_v^{\mathrm{sys}}(w_v)$ and $c_v^{\tau}(w_v)$ equal the energy cost and the latency cost owing to the wireless transmission.

Then, finding an optimal solution in P1 is equivalent to finding an optimal routing strategy in $\overline{\mathcal{G}}$ with the minimum routing cost. The task flows passing through different vertices represent different strategies for offloading. We rewrite the objective function, $\Phi\left(\boldsymbol{X}, \boldsymbol{Y}\right)$, as

$$\begin{aligned} \Phi\left(\boldsymbol{X}, \boldsymbol{Y}\right) &= \sum_{v \in \overline{\mathcal{V}}} w_v c_v^{\mathrm{sys}}(w_v) + \sum_{v \in \overline{\mathcal{V}}} w_v c_v^{\tau}(w_v) \\ &\quad + \sum_{e \in \overline{\mathcal{E}}} w_e c_e^{\tau}(w_e) + \sum_{v \in \mathcal{V}^{\mathrm{mec}}} \gamma_d \gamma_e \alpha E_v^{\max}. \qquad (16) \end{aligned}$$

Here, we use an example to make the content clearer. We use a network instance in Figure 2(a) which consists of two UE (i.e., UE$_1$ and UE$_2$), three APs (i.e., F$_A$, F$_B$, F$_C$) and a data center (i.e., M$_C$). UE$_1$ is covered by both F$_A$ and F$_C$ while UE$_2$ can only connect to F$_C$. In APs, F$_B$ is a pure AP while F$_A$ and F$_C$ are APs equipped with clouds (M$_A$ and M$_C$, respectively). F$_D$ is a forwarding unit (e.g., switch) in the core network, connecting F$_A$, F$_B$ and M$_B$. The corresponding extended graph, $\overline{\mathcal{G}}$, is shown in Figure 2(b). We assume that each UE has three different transmission power levels.

$$U_{u,f}^{\mathrm{mar}}(s) = \sum_{v\in\mathcal{V}^{\mathrm{mec}}\cap f}\left[\gamma_d\gamma_e(1-\alpha)E_v^{\max}\frac{\kappa}{K_v^{\mathrm{mec}}} + (1-\gamma_d)\left(2\frac{\xi_{j,3}\xi_{j,4}w_j}{\left(K_j^{\mathrm{mec}}\right)^2} + \frac{\xi_{j,3}}{K_j^{\mathrm{mec}}}\right)\right] + \sum_{v\in\mathcal{V}^{\mathrm{rou}}\cap f}(1-\gamma_d)\left(2\xi_{v,1}w_v + \xi_{v,2}\right)$$
$$+ \sum_{v\in\mathcal{V}^{\mathrm{lc}}\cap f}\left((1-\gamma_e)\gamma_d\varsigma_u\kappa K_{u,0}^2 + (1-\gamma_d)\frac{\kappa}{K_{u,0}}\right) + \sum_{v\in\mathcal{V}^{\mathrm{p}}\cap f}\left(\frac{\gamma_d(1-\gamma_e)p_{u,m} + (1-\gamma_d)}{K_{u,m}}\right) + \sum_{e\in\mathcal{E}\cap f}\frac{1-\gamma_d}{K_e^{\mathrm{link}}}. \quad (17)$$

We focus on the offloading strategy of UE$_1$. In $\overline{\mathcal{G}}$, UE$_1$ needs to route its demands (tasks) to the v^{dest}. Two feasible paths (the dashed lines in blue) are shown in Figure 2 (b). The path on the left (passing through L_A) represents the local processing manner. The path $\{S_1, P_A, F_A, F_B, F_C, M_B, v^{\mathrm{dest}}\}$ (on the right side) represents an offloading strategy. This strategy implies that UE$_1$ firstly transmits the tasks to F_A with power level P_A. Then, the network routes the tasks along the path $\{F_B, F_C, M_B\}$ and deliver the tasks to the data center M_B. It can be seen that the path selection of UE affects $\Phi(\boldsymbol{X}, \boldsymbol{Y})$ through changing the network congestion.

We extend $\mathcal{F}$ to represent the feasible paths in $\overline{\mathcal{G}}$. The flow in $\overline{\mathcal{G}}$ is denoted as $s = \{s_u\}_{u\in\mathcal{U}}$, where $s_u = \{s_{u,f}\}_{f\in\mathcal{F}}$. Similarly, $s_{u,f}$ is the portion of tasks which come from u and choose the path f. For simplicity, we represent $\Phi(\boldsymbol{X}, \boldsymbol{Y})$ in (16) as a function regarding s, i.e. $\Phi(s)$.

B. THE PROPOSED ENERGY-LATENCY AWARE OFFLOADING GAME

We defined an offloading game, denoted as $G = \{\mathcal{U}, \mathcal{S}, \boldsymbol{U}(s)\}$, where $\mathcal{U}$, $\mathcal{S}$, and $\boldsymbol{U}(s)$ are the player set, the strategy space, and the payoff function, respectively. $\mathcal{U}$ consists of UE which has a demand $r_u = \omega_u L$. The strategy space contains all the feasible flow in $\overline{\mathcal{G}}$. We design the proposed game with the marginal payoff function. The marginal payoff function represents additional utility caused by a new player when the other players' actions are given. In our game, the payoff function is defined as $\boldsymbol{U}(s) = \{U_{u,f}^{\max}(s)\}_{u\in\mathcal{U}, f\in\mathcal{F}}$, where $U_{u,f}^{\max}(s)$ is represented in (17), as shown at the top of this page. $U_{u,f}^{\mathrm{mar}}(s)$ consists of two major terms: 1) the power pricing consisting of additional energy consumption when u chooses f; 2) the latency-aware utility motivated by additional response time.

In the proposed game, the players select the feasible flow with the minimum marginal cost. We use the best response dynamics to describe how s evolves over time. The differential inclusion describes the tendency of s_u, which is represented as

$$\dot{s}_u \in V_u(s) = B_u(s) - s_u, \quad (18)$$

where $B_u(s)$ is the mixed best response correspondence [49].

C. THE PROPERTY OF THE GAME EQUILIBRIUM

In this section, we analyze equilibrium properties of the offloading game, including the convergence and the efficiency.

Theorem 1: The game with the marginal payoff function defined in (17) *is a potential game. The potential function is* $\phi(s) = \Phi(s) - \sum_{v\in\mathcal{V}^{\mathrm{mec}}}\gamma_d\gamma_r\alpha E_v^{\max}$.

Proof: The nonatomic routing games are indeed potential games with the potential function defined as

$$\phi(s) = \sum_{v\in\overline{\mathcal{V}}}\int_0^{w_v} c_v^{\mathrm{r}}(z)\,dz + \sum_{e\in\overline{\mathcal{E}}}\int_0^{w_e} c_e^{\mathrm{r}}(z)\,dz, \quad (19)$$

where $c_v^{\mathrm{r}}(\cdot)$ and $c_e^{\mathrm{r}}(\cdot)$ is the cost motivated by the payoff function on $v \in \overline{\mathcal{V}}$ and $e \in \overline{\mathcal{E}}$ [44]. In our offloading game, $c_e^{\mathrm{r}}(w_e) = \frac{1-\gamma_d}{K_e^{\mathrm{link}}}\delta_e^{\mathcal{E}}$ and $c_v^{\mathrm{r}}(w_v)$ is represented in (20), as shown at the top of the next page.

Therefore, to prove Theorem 1, it is sufficient to check whether $\phi(s) = \Phi(s) - \sum_{v\in\mathcal{V}^{\mathrm{mec}}}\gamma_d\gamma_e\alpha E_v^{\max}$. We have

$$\int_0^{w_v} c_v^{\mathrm{r}}(z)dz = w_v c_v^{\mathrm{sys}}(w_v) + w_v c_v^{\tau}(w_v), \quad \forall v \in \overline{\mathcal{V}}, \quad (21)$$

and

$$\int_0^{w_e} c_e^{\mathrm{r}}(z)dz = w_e c_e^{\tau}(w_e), \forall e \in \overline{\mathcal{E}}. \quad (22)$$

By summing up all terms in $\overline{\mathcal{V}}$ and $\overline{\mathcal{E}}$, we can deduce $\phi(s) = \Phi(s) - \sum_{v\in\mathcal{V}^{\mathrm{mec}}}\gamma_d\gamma_e\alpha E_v^{\max}$. □

Theorem 2: The equilibrium of the proposed game, $\overline{s}$, *is the globally optimal solution regarding* $\Phi(s)$.

Proof: We have shown that the proposed game is a potential game. Since $\overline{s}$ is the equilibrium of the game, we can deduce that $\overline{s}$ satisfies the Karush-Kuhn-Tucher (KKT) conditions of the problem which locally minimizes the potential function $\phi(s)$ [49]. Now, we prove the local minimizer of $\phi(s)$ is a globally optimal solution of $\Phi(s)$. First, $w_v c_v^{\mathrm{sys}}(w_v)$, $w_v c_v^{\tau}(w_v)$, and $w_e c_e^{\tau}(w_e)$ are convex functions regarding w_v and w_e. Meanwhile, w_v and w_e are linear combinations of s. Then, $\phi(s)$ is a convex function regarding s. Thus, we can deduce that $\overline{s}$ is the globally optimal solution of $\phi(s)$. Since $\Phi(s) = \phi(s) + \sum_{v\in\mathcal{V}^{\mathrm{mec}}}\gamma_d\gamma_e\alpha E_v^{\max}$ and $\sum_{v\in\mathcal{V}^{\mathrm{mec}}}\gamma_d\gamma_e\alpha E_v^{\max}$ is a constant, we can conclude that $\overline{s}$ is a globally optimal solution of $\Phi(s)$. □

According to the definition, $\overline{s}$ is a mixed strategy Nash equilibrium. The existence of $\overline{s}$ is guaranteed since the mixed strategy Nash equilibrium always exists [49]. The convergence of the game under the best response dynamics (18) could also be ensured. The reason is that the best response dynamic satisfies the properties of the positive correlation (PC) and the Nash stationarity (NS) [49]. The PC and NS guarantee that the equilibrium of the game coincides

$$c_v^{\mathrm{r}}(w_v) = (1-\gamma_d)\left(2\xi_{v,1}w_v+\xi_{v,2}\right)\delta_v^{\mathrm{rou}} + \left[\gamma_d\gamma_e(1-\alpha)E_v^{\max}\frac{\kappa}{K_v^{\mathrm{mec}}} + (1-\gamma_d)\left(2\frac{\xi_{j,3}\xi_{v,4}w_v}{\left(K_v^{\mathrm{mec}}\right)^2}+\frac{\xi_{v,3}}{K_v^{\mathrm{mec}}}\right)\right]\delta_v^{\mathrm{mec}} + \left[(1-\gamma_e)\gamma_d\varsigma_u\kappa K_{u,0}^2 + (1-\gamma_d)\frac{\kappa}{K_{u,0}}\right]\delta_v^{\mathrm{lc}} + \frac{\gamma_d(1-\gamma_e)p_{u,m}+(1-\gamma_d)}{K_{u,m}}\delta_v^{\mathrm{p}} \tag{20}$$

with the stationary point of (18). Thus, the behaviors of players under the best response dynamics eventually lead to the equilibrium of the game. The detail information can be found in [49].

D. THE PROPOSED GAME THEORY BASED ALGORITHM

We have shown the existence and the convergence of the game equilibrium. Meanwhile, Theorem 2 has shown that the behaviors guided by the marginal payoff function result in global improvement. Thus, we propose an energy-latency aware offloading algorithm based on the proposed offloading game, which is shown in Algorithm 1.

Algorithm 1 Best Response-Based Offloading Algorithm (BROA)

```
Each AP $b \in \mathcal{B}$ :
while True do
    Collect the information for $U_{b,f}^{\mathrm{ap}}(s)$
    Compute $U_{b,f}^{\mathrm{ap}}(s), \forall f \in \{f : b \in f\}$
    Broadcast $U_{b,f}^{\mathrm{ap}}(s)$
end
Each UE $u \in \mathcal{U}$ :
Initialize virtual tasks $\mathcal{I}$
Assign feasible routes for $\mathcal{I}$ randomly
Report $s_u$ to the network
while Not converged do
    for Each $i \in \mathcal{I}$ do
        Access the information from AP $b \in \mathcal{B}$
        Calculate $U_{u,f}^{\mathrm{mar}}(s), \forall f \in \mathcal{F}_u$
        Set $f^* = \arg\ \min_f U_{u,f}^{\mathrm{mar}}(s)$
        if $U_{u,f^*}^{mar}(s) \le U_{u,s_i^{vir}}^{mar}(s)$ then
            Update $s_i^{\mathrm{vir}}$, $s_u$ and report $s_u$ to the network
        end
    end
end
```

In Algorithm 1, players collaborate with each other to find the equilibrium of the offloading game. Usually, it is hard to find a mixed strategy Nash equilibrium with high-dimensional action space [50]. Thus, we use the atomic routing to approximate the equilibrium of G. For each player, a set of virtual tasks ($\mathcal{I}_u$) is introduced and $\mathcal{I} = \cup_u \mathcal{I}_u$. The length of virtual tasks in $\mathcal{I}_u$ is initialized as $r_i^{\mathrm{vir}} = \omega_u L/|\mathcal{I}_u|$. In order to ensure the optimality of the results, we set $|\mathcal{I}_u| \ge \sqrt{|\mathcal{F}_u|}, \forall u \in \mathcal{U}$ [46]. During each iteration in BROA, $i \in \mathcal{I}$ picks a feasible path $s_i^{\mathrm{vir}} \in \mathcal{F}$ with minimum cost. We use $s^{\mathrm{vir}} = \{s_i^{\mathrm{vir}}\}_{i\in\mathcal{I}}$ to denote the joint strategies of virtual tasks. When s^{vir} converges, $\bar{s}$ could be obtained. Then, UE can deduce $\boldsymbol{X}$ and $\boldsymbol{Y}$ based on $\bar{s}$.

To improve the scalability of the network, BROA works in a distributed manner. We introduce two roles, i.e., APs and UE. Each AP, $b \in \mathcal{B}$, calculates $U_{b,f}^{\mathrm{ap}}(s), \forall f \in \{f : b \in f\}$ in (23) and broadcasts the results in its coverage.

$$U_{b,f}^{\mathrm{ap}}(s) = \sum_{v\in\mathcal{V}^{\mathrm{mec}}\cap f}\gamma_d\gamma_e(1-\alpha)E_v^{\max}\frac{\kappa}{K_v^{\mathrm{mec}}} + \sum_{e\in\mathcal{E}\cap f}\frac{1-\gamma_d}{K_e^{\mathrm{link}}} + \sum_{v\in\mathcal{V}^{\mathrm{rou}}\cap f}(1-\gamma_d)\left(2\xi_{v,1}w_v+\xi_{v,2}\right) + \sum_{v\in\mathcal{V}^{\mathrm{mec}}\cap f}(1-\gamma_d)\left(2\frac{\xi_{j,3}\xi_{j,4}w_j}{\left(K_j^{\mathrm{mec}}\right)^2}+\frac{\xi_{j,3}}{K_j^{\mathrm{mec}}}\right). \tag{23}$$

Once UE receives the information, including $U_{b,f}^{\mathrm{ap}}(s)$ and channel coefficients, $U_{u,f}^{\mathrm{mar}}(s)$ can be calculated. Then, each UE sequentially revisits the virtual tasks' strategies, which in return changes the network congestion. A stable network state is reached when all virtual tasks adopt their optimal paths. It can be observed that the calculations in BROA are carried out in a distributed manner, which is a preferred feature in a large scale network.

V. IMPROVEMENT BASED ON THE APPROXIMATE FACTOR

In this section, we propose an approximate algorithm FCOA to accelerate the convergence speed of BROA. A long convergence time of BROA results in additional costs, e.g., signal overheads, energy consumption. We evaluate the convergence speed in terms of overall re-routing times. In this paper, re-routing means a path switching (or strategy revisiting) of virtual tasks.

The proposed FCOA is detailed in Algorithm 2, in which we introduce an approximate factor β ($\beta \ge 1$). The approximate factor builds a barrier for the path switching. Virtual tasks would stick to the current paths when the cost-saving (of a new path) is tiny.

In order to track the performance of FCOA, we define the game $G^{\mathrm{vir}} = \{\mathcal{I}, \mathcal{F}, U^{\mathrm{vir}}(s^{\mathrm{vir}})\}$, in which players are virtual tasks. Each player i chooses its strategy s_i^{vir} from feasible paths in $\mathcal{F}$. Here, we emphasize that s_i^{vir} is a path. The current strategy profile of $\mathcal{I}$ is denoted as $s^{\mathrm{vir}} = \{s_i^{\mathrm{vir}}\}_{i\in\mathcal{I}}$. The payoff

Algorithm 2 The Fast-Converged Offloading Algorithm (FCOA)

```
Each AP $b \in \mathcal{B}$ :
while True do
    Collect the information for $U^{\mathrm{ap}}_{b,f}(\boldsymbol{s})$
    Compute $U^{\mathrm{ap}}_{b,f}(s)$, $\forall f \in \{f : b \in f\}$
    Broadcast $U^{\mathrm{ap}}_{b,f}(s)$
end
Each UE $u \in \mathcal{U}$ :
Initialize virtual tasks $\mathcal{I}$
Assign feasible routes for $\mathcal{I}$ randomly
Report $s_u$ to the network
while Not converged do
    for Each $i \in \mathcal{I}$ do
        Access the information from AP $b \in \mathcal{B}$
        Calculate $U^{\mathrm{mar}}_{u,f}(s)$, $\forall f \in \mathcal{F}_u$
        Set $f^* = \arg\ \min_f U^{\mathrm{mar}}_{u,f}(s)$
        if $\beta \cdot U^{mar}_{u,f^*}(s) \le U^{mar}_{u,s^{vir}_i}(s)$ then
            Update $s^{\mathrm{vir}}_i$, $s_u$ and report $s_u$ to the network
        end
    end
end
```

function in G^{vir} is rewritten as

$$U^{\mathrm{vir}}_i(s^{\mathrm{vir}}_i, s^{\mathrm{vir}}_{-i}) = r^{\mathrm{vir}}_i \left[\sum_{v \in \overline{\mathcal{V}} \cap s^{\mathrm{vir}}_i} c^{\mathrm{r}}_v\left(w_v\right) + \sum_{e \in \overline{\mathcal{E}} \cap s^{\mathrm{vir}}_i} c^{\mathrm{r}}_e\left(w_e\right) \right] \quad (24)$$

We adopt β-improving deviation in FCOA. A virtual task i switches to another path $\widetilde{s}^{\mathrm{vir}}_i$ if and only if (25) holds.

$$\beta \cdot U^{\mathrm{vir}}_i\left(\widetilde{s}^{\mathrm{vir}}_i, s^{\mathrm{vir}}_{-i}\right) \le U^{\mathrm{vir}}_i\left(s^{\mathrm{vir}}_i, s^{\mathrm{vir}}_{-i}\right) \quad (25)$$

We define the social cost of G^{vir} as

$$\begin{aligned} C(s^{\mathrm{vir}}) &= \sum_{i \in \mathcal{I}} U^{\mathrm{vir}}_i(s^{\mathrm{vir}}) \\ &= \sum_{v \in \overline{\mathcal{V}}} w_v c^{\mathrm{r}}_v\left(w_v\right) + \sum_{e \in \overline{\mathcal{E}}} w_e c^{\mathrm{r}}_e\left(w_e\right). \end{aligned} \quad (26)$$

Theorem 3: The FCOA algorithm converges with any initial state s^{vir}_{init}.

Proof: In order to prove the convergence, we introduce an auxiliary function $g(s^{\mathrm{vir}})$ represented as

$$g\left(s^{\mathrm{vir}}\right) = C\left(s^{\mathrm{vir}}\right) + W\left(s^{\mathrm{vir}}\right), \quad (27)$$

where

$$\begin{aligned} W\left(s^{\mathrm{vir}}\right) &= \sum_{i \in \mathcal{I}} \sum_{v \in \overline{\mathcal{V}} \cap s^{\mathrm{vir}}_i} r^{\mathrm{vir}}_i c^{\mathrm{r}}_v\left(r^{\mathrm{vir}}_i\right) \\ &\quad + \sum_{i \in \mathcal{I}} \sum_{v \in \overline{\mathcal{V}} \cap s^{\mathrm{vir}}_i} r^{\mathrm{vir}}_i c^{\mathrm{r}}_e\left(r^{\mathrm{vir}}_i\right). \end{aligned} \quad (28)$$

Suppose a virtual task $i \in \mathcal{I}$ re-routes to $\widetilde{s}^{\mathrm{vir}}_i$ when it currently adopts s^{vir}_i. We can deduce that

$$\begin{aligned} \Delta g(s^{\mathrm{vir}}_i, s^{\mathrm{vir}}_{-i}) &= g(\widetilde{s}^{\mathrm{vir}}_i, s^{\mathrm{vir}}_{-i}) - g(s^{\mathrm{vir}}_i, s^{\mathrm{vir}}_{-i}) \\ &\overset{(a)}{=} 2 r^{\mathrm{vir}}_i \left[U^{\mathrm{vir}}_i\left(\widetilde{s}^{\mathrm{vir}}_i, s^{\mathrm{vir}}_{-i}\right) - U^{\mathrm{vir}}_i\left(s^{\mathrm{vir}}_i, s^{\mathrm{vir}}_{-i}\right) \right] \\ &\overset{(b)}{<} 0, \end{aligned} \quad (29)$$

where (a) holds since $U^{\mathrm{vir}}_i(\cdot)$ is an affine function [45]. (b) holds due to the β-improving deviation. (29) shows that the re-routing of virtual tasks in FCOA monotonously decreases $g(s^{\mathrm{vir}})$ from any initial state. We use $g(s^{\mathrm{vir}})$ as the Lyapunov function. Since $g(s^{\mathrm{vir}})$ is lower bounded (i.e., $g(s^{\mathrm{vir}}) \ge 0$), we can conclude that FCOA converges from any initial state according to Lyapunov stability theorem. □

Theorem 4: FCOA converges to an equilibrium $\overline{s}^{vir}$ from s^{vir}_{init} in $O\left(\frac{1}{\delta\left(\min_i\{r^{vir}_i\}\right)^2}\left(\frac{\beta}{\beta-1}\right)\log\left(\frac{g\left(s^{vir}_{init}\right)}{g\left(\overline{s}^{vir}\right)}\right)\right)$ times of re-routing, where δ is a constant represented in (30)*, as shown at the bottom of this page.*

Proof: First, we can have the following inequation due to the definition of $W(s^{\mathrm{vir}})$,

$$\begin{aligned} W\left(s^{\mathrm{vir}}\right) &= \sum_{i \in \mathcal{I}} r^{\mathrm{vir}}_i \left[\sum_{v \in \overline{\mathcal{V}} \cap s^{\mathrm{vir}}_i} c^{\mathrm{r}}_v\left(r^{\mathrm{vir}}_i\right) + \sum_{e \in \overline{\mathcal{E}} \cap s^{\mathrm{vir}}_i} c^{\mathrm{r}}_e\left(r^{\mathrm{vir}}_i\right) \right] \\ &\le \sum_{i \in \mathcal{I}} U^{\mathrm{vir}}_i\left(s^{\mathrm{vir}}\right) \le C\left(s^{\mathrm{vir}}\right). \end{aligned} \quad (31)$$

Thus, we have

$$C\left(s^{\mathrm{vir}}\right) \ge \frac{1}{2} g\left(s^{\mathrm{vir}}\right). \quad (32)$$

Suppose a virtual task $i \in \mathcal{I}$ applies the β-improving deviation from s^{vir} to $\widetilde{s}^{\mathrm{vir}}$. We have

$$\begin{aligned} \Delta_i\left(s^{\mathrm{vir}}\right) &= U^{\mathrm{vir}}_i\left(s^{\mathrm{vir}}\right) - U^{\mathrm{vir}}_i\left(\widetilde{s}^{\mathrm{vir}}\right) \\ &\ge \left(1 - \frac{1}{\beta}\right) U^{\mathrm{vir}}_i\left(s^{\mathrm{vir}}\right). \end{aligned} \quad (33)$$

Additionally, we can deduce

$$\frac{U^{\mathrm{vir}}_i\left(s^{\mathrm{vir}}\right)}{C\left(s^{\mathrm{vir}}\right)} \ge \min_i\{r^{\mathrm{vir}}_i\} \cdot \delta, \quad \forall i \in \mathcal{I}. \quad (34)$$

$$\delta = \frac{\min\left\{ \min_{v \in \mathcal{V}^{\mathrm{mec}}} \left\{ \gamma_d \gamma_e \left(1 - \alpha\right) E^{\mathrm{max}}_v \frac{\kappa}{K^{\mathrm{mec}}_v} + \left(1 - \gamma_d\right) \frac{\xi_{j,3}}{K^{\mathrm{mec}}_j} \right\}, \min_{v \in \mathcal{V}^{\mathrm{lp}}} \left\{ \left(1 - \gamma_e\right) \gamma_d \varsigma_u \kappa K^2_{u,0} + \left(1 - \gamma_d\right) \frac{\kappa}{K_{u,0}} \right\} \right\}}{\sum_{v \in \overline{\mathcal{V}}} \left[B^{\mathrm{bw}}_v \cdot c^{\mathrm{r}}_v\left(B^{\mathrm{bw}}_v\right) \right] + \sum_{e \in \overline{\mathcal{E}}} \left[B^{\mathrm{bw}}_e \cdot c^{\mathrm{r}}_e\left(B^{\mathrm{bw}}_e\right) \right]} \quad (30)$$

Combined with $\Delta g\left(s^{\text{vir}}\right) = 2\, r_i^{\text{vir}} \Delta_i\left(s^{\text{vir}}\right)$ [45], we have

$$\begin{aligned}\Delta g\left(s^{\text{vir}}\right) &\geq 2\min\left\{r_i^{\text{vir}}\right\}\Delta_i\left(s^{\text{vir}}\right)\\ &\geq 2\min\left\{r_i^{\text{vir}}\right\}^2\delta\left(1-\frac{1}{\beta}\right)C\left(s^{\text{vir}}\right)\\ &\geq \min\left\{r_i^{\text{vir}}\right\}^2\delta\left(1-\frac{1}{\beta}\right)g\left(s^{\text{vir}}\right). \end{aligned} \tag{35}$$

Therefore, the value of auxiliary function $g(\cdot)$ at $\overline{s}^{\text{vir}}$ can be represented as

$$g\left(\overline{s}^{\text{vir}}\right) \leq [1-Q]^N g\left(s_{\text{vir}}^{\text{init}}\right), \tag{36}$$

where $Q = \left(\min\left\{r_i^{\text{vir}}\right\}\right)^2\delta\left(1-\frac{1}{\beta}\right)$. From the equation above, the upper bound of steps follows by (37).

$$O\left(\frac{1}{\delta\left(\min\left\{r_i^{\text{vir}}\right\}\right)^2}\left(\frac{\beta}{\beta-1}\right)\log\left(\frac{g\left(s_{\text{init}}^{\text{vir}}\right)}{g\left(\overline{s}^{\text{vir}}\right)}\right)\right). \tag{37}$$

□

Theorem 5: The efficiency of the FCOA algorithm admits at least

$$\frac{\phi\left(\overline{s}^{vir}\right)}{\phi\left(\hat{s}^{vir}\right)} \leq \left(3+\sqrt{5}\right)\beta, \tag{38}$$

where $\overline{s}^{vir}$ and $\widehat{s}^{vir}$ denote the equilibrium of G^{vir} and the globally optimal solution, respectively.

Proof: Since $c_v^{\text{r}}\left(w_e\right)$ and $c_e^{\text{r}}\left(w_e\right)$ are linear functions, we re-write them as $c_v^{\text{r}}\left(w_v\right) = a_v w_v + b_v$ and $c_e^{\text{r}}\left(w_e\right) = a_e w_e + b_e$, where

$$a_v = 2\left(1-\gamma_d\right)\xi_{v,1}\delta_v^{\text{rou}} + 2\left(1-\gamma_d\right)\frac{\xi_{v,3}\xi_{v,4}}{\left(K_v^{\text{mec}}\right)^2}\delta_v^{\text{mec}}, \tag{39}$$

$$\begin{aligned} b_v = &\left(1-\gamma_d\right)\xi_{v,2}\delta_v^{\text{rou}}\\ &+\left[\gamma_d\gamma_e\left(1-\alpha\right)E_v^{\max}\frac{\kappa}{K_v^{\text{mec}}} + \left(1-\gamma_d\right)\frac{\xi_{v,3}}{K_v^{\text{mec}}}\right]\delta_v^{\text{mec}}\\ &+\left[\left(1-\gamma_e\right)\gamma_d\varsigma_u\kappa K_{u,0}^2 + \left(1-\gamma_d\right)\frac{\kappa}{K_{u,0}}\right]\delta_v^{\text{lc}}\\ &+\frac{\gamma_d\left(1-\gamma_e\right)p_{u,m} + \left(1-\gamma_d\right)}{K_{u,m}}\delta_v^{\text{p}}, \end{aligned} \tag{40}$$

and

$$a_e = 0, \qquad b_e = \frac{(1-\gamma_d)\delta_e^{\mathcal{E}}}{K_e^{\text{link}}}. \tag{41}$$

By the definition of $C\left(s^{\text{vir}}\right)$ and $\phi\left(s^{\text{vir}}\right)$, we can deduce

$$\phi\left(s^{\text{vir}}\right) \leq C\left(s^{\text{vir}}\right) \leq 2\phi\left(s^{\text{vir}}\right). \tag{42}$$

Since a virtual task i would not switch to $\widehat{s}^{\text{vir}}$ when it chooses $\overline{s}^{\text{vir}}$, we have

$$\begin{aligned}&\sum_{v\in\overline{\mathcal{V}}\cap\overline{s}^{\text{vir}}} c_v^{\text{r}}\left(\overline{w}_v\right) + \sum_{e\in\overline{\mathcal{E}}\cap\overline{s}^{\text{vir}}} c_e^{\text{r}}\left(\overline{w}_e\right)\\ &\leq \beta\left[\sum_{v\in\overline{\mathcal{V}}\cap\widehat{s}^{\text{vir}}} c_v^{\text{r}}\left(\overline{w}_v + r_i^{\text{vir}}\right) + \sum_{e\in\overline{\mathcal{E}}\cap\widehat{s}^{\text{vir}}} c_e^{\text{r}}\left(\overline{w}_e + r_i^{\text{vir}}\right)\right],\end{aligned} \tag{43}$$

where $\overline{w}_v$ and $\overline{w}_e$ is the aggregate traffic loads on v and e at $\overline{s}^{\text{vir}}$. We multiply the inequality by r_i^{vir} for each i. Summing up the resulting $|\mathcal{I}|$ inequalities, we obtain

$$\begin{aligned}C\left(\overline{s}^{\text{vir}}\right) &\leq \beta\sum_{i\in\mathcal{I}} r_i^{\text{vir}} \sum_{v\in\overline{\mathcal{V}}\cap\hat{s}^{\text{vir}}}\left[a_v\left(\overline{w}_v + r_i^{\text{vir}}\right) + b_v\right]\\ &\quad+\beta\sum_{i\in\mathcal{I}} r_i^{\text{vir}} \sum_{e\in\overline{\mathcal{E}}\cap\hat{s}^{\text{vir}}}\left[a_e\left(\overline{w}_e + r_i^{\text{vir}}\right) + b_e\right]\\ &\leq \beta\sum_{i\in\mathcal{I}} r_i^{\text{vir}} \sum_{v\in\overline{\mathcal{V}}\cap\hat{s}^{\text{vir}}}\left[a_v\left(\overline{w}_v + \hat{w}_v\right) + b_v\right]\\ &\quad+\beta\sum_{i\in\mathcal{I}} r_i^{\text{vir}} \sum_{e\in\overline{\mathcal{E}}\cap\hat{s}^{\text{vir}}}\left[a_e\left(\overline{w}_e + \hat{w}_v\right) + b_e\right]\\ &\leq \beta\sum_{v\in\overline{\mathcal{V}}}\left[a_v\left(\overline{w}_v + \hat{w}_v\right) + b_v\right]\hat{w}_v\\ &\quad+\beta\sum_{e\in\overline{\mathcal{E}}}\left[a_e\left(\overline{w}_e + \hat{w}_v\right) + b_e\right]\hat{w}_e\\ &= \beta\cdot C\left(\widehat{s}^{\text{vir}}\right) + \beta\left(\sum_{v\in\overline{\mathcal{V}}} a_v\overline{w}_v\hat{w}_v + \sum_{e\in\overline{\mathcal{E}}} a_e\overline{w}_e\hat{w}_e\right),\end{aligned} \tag{44}$$

where $\hat{w}_v$ and $\hat{w}_e$ is the aggregate traffic loads on v and e at $\hat{s}^{\text{vir}}$.

Also, we have

$$\begin{aligned}&\sum_{v\in\overline{\mathcal{V}}} a_v\overline{w}_v\widehat{w}_v + \sum_{e\in\overline{\mathcal{E}}} a_e\overline{w}_e\widehat{w}_e\\ &\leq \sqrt{\sum_{v\in\overline{\mathcal{V}}} a_v\overline{w}_v^{\,2}}\sqrt{\sum_{v\in\overline{\mathcal{V}}} a_v\widehat{w}_v^{\,2}} + \sqrt{\sum_{e\in\overline{\mathcal{E}}} a_e\overline{w}_e^{\,2}}\sqrt{\sum_{v\in\overline{\mathcal{E}}} a_e\widehat{w}_e^{\,2}}\\ &\leq \sqrt{C\left(\overline{s}^{\text{vir}}\right)}\sqrt{C\left(\widehat{s}^{vir}\right)}.\end{aligned} \tag{45}$$

Substituting (45) into (44), we can deduce

$$\begin{aligned}C\left(\overline{s}^{\text{vir}}\right) &\leq \beta C\left(\widehat{s}^{\text{vir}}\right) + \beta\sqrt{C\left(\overline{s}^{\text{vir}}\right)}\sqrt{C\left(\widehat{s}^{\text{vir}}\right)}\\ \frac{C\left(\overline{s}^{\text{vir}}\right)}{C\left(\widehat{s}^{\text{vir}}\right)} - \beta &\leq \beta\frac{\sqrt{C\left(\overline{s}^{\text{vir}}\right)}}{\sqrt{C\left(\widehat{s}^{\text{vir}}\right)}}.\end{aligned} \tag{46}$$

By solving (46), we have

$$\frac{C\left(\overline{s}^{\text{vir}}\right)}{C\left(\widehat{s}^{\text{vir}}\right)} \leq \frac{3\beta + \sqrt{5}\beta}{2}. \tag{47}$$

Finally, we have

$$\frac{\phi\left(\overline{s}^{\text{vir}}\right)}{\phi\left(\hat{s}^{\text{vir}}\right)} \leq \left(3+\sqrt{5}\right)\beta. \tag{48}$$

□

VI. NUMERICAL RESULTS AND ANALYSIS

We consider a hierarchical MEC network in Figure 1. In this network, we deploy a number of APs covering an area of 0.81 km^2. The aggregate nodes and core nodes are set according to part of Atlanta's core network [12]. The latency on the routers is configured as follows to cope for different scopes: 1∼2 ms per task for the aggregation layer and 2∼5 ms per task for the core layer. The capacities of the wired links are set as 10 Gbps, uniformly. We install an edge cloud for each AP and set up two central clouds on the core nodes. We manage the resources on the clouds in the form of VMs. Each VM is allocated with one CPU core. We list the physical configurations of the clouds in Table 3. The parameters are designed according to [27].

TABLE 3. **Configuration for the clouds.**

	Edge clouds	Central clouds
α	0.2	0.2
$\xi_{j,3}$	3000	4300
$\xi_{j,4}$	1500	1500
$E_j^{\max}$	0.64 W	1.28 W
CPU frequency (K_j^{mec})	0.3 GHz	0.8 GHz
Number of CPU cores	4	8

The UE is randomly dispersed in the area. The physical profile of UE is listed in Table 4. The uplink channel gains are generated using a distance-dependent path loss model given as $L = 140.7 + 36.7\log_{10}(d/1000)$ [51] in dB. Each UE runs an MEC application with the task length equaling 5 Kb. κ is 1500 CPU cycles per bit on default. The UE generates the task at the speed of 10 or 15 per second, randomly. In both BROA and FCOA, we set $|\mathcal{I}_u| = 30$ on default.

TABLE 4. **Configuration for the UE.**

	UE
$K_{u,0}$	0.1 GHz
ς	0.2×10^{-25}
σ^2	-90 dBm
B	0.2 MHz
Γ	0.9
$P_{u,m}$	[10 dBm,15 dBm,20 dBm,25 dBm]

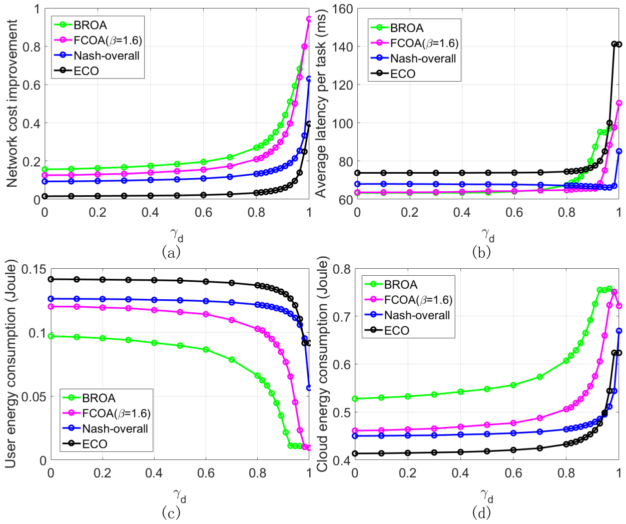
We evaluate the algorithm performance using the Mento-Carlo method. We compare the proposed algorithms with the following baselines.

- Local processing (LP): LP processes the tasks in a local manner. No offloading decision, the uplink transmission power control, the cloud selection, and the task routing are involved.
- Edge clouds only (ECO): UE only considers the edge clouds under this algorithm. The offloading decision, the uplink transmission power control, and the cloud selection are involved in ECO.
- Nash-overall: Both edge clouds and central clouds are taken into consideration. This algorithm solves the optimization problem based on the congestion game, which is proposed in [52]. In this algorithm, UE selfishly chooses the path in a greedy manner.

The comparison to LP gives us the offloading gain (network cost improvement) of our proposed algorithms. We use ECO as a baseline since it can demonstrate the advantages of the hierarchical MEC architecture. The comparison to Nash-overall emphasizes the benefits of our marginal payoff function.

We first validate the algorithm performance with different weighted factors in (12), i.e., γ_d and γ_e.

Figure 3 discusses the impacts of γ_d. In order to quantitatively assess the advantages, we show the network cost improvement compared to LP in Figure 3(a). Clearly, BROA dominates the others over the entire region of γ_d. When γ_d equals 1, BROA can decrease the network cost by 94.27% compared to LP. FCOA also outperforms the conventional algorithms (e.g., Nash-overall and ECO). It dominates LP by at least 15.6%. Figure 3(b) verifies that the task latency in BROA and FCOA monotonously increases with γ_d. Also, it can be seen that, among all the algorithms, BROA is more sensitive on γ_d. The task latency in BROA increases when γ_d is larger than 0.6. However, the other algorithms (FCOA, Nash-overall, ECO) start to react to γ_d until γ_d is above 0.8. The reason is that BROA can track the global optima in more accurate manner. Each iteration in BROA can globally improve the network cost.



(a)

(b)

(c)

(d)

FIGURE 3. **Algorithm performance under different γ_d with $|\mathcal{U}| = 60$, $|\mathcal{B}| = 2$, and $\gamma_e = 0.3$.**

Figure 3(c) and (d) tell the energy consumption on the UE and the clouds, respectively. Take into account these figures, the offloading ratio in different algorithms can also be deduced. In our simulation, the clouds are more energy efficient (about 0.15 Joule per GHz on the central clouds, 0.25 Joule per GHz on the edge clouds) than the local computing manner (2.0 Joule per GHz). Thus, as γ_d increases, more tasks are offloaded to save energy, which is illustrated in Figure 3(c). When γ_d is above 0.9, nearly all tasks are uploaded in BROA and FCOA. To further reduce the energy consumption, UE begins to decrease their transmission power. Thus, a sharp point appears at $\gamma_d = 0.92$ in BROA ($\gamma_d = 0.98$ in FCOA). Figure 3(d) elaborates on how the tasks migrate on the network side. We can see that, at the beginning, the energy consumption on clouds continuously grows. When γ_d is above 0.96 in BROA (0.98 in FCOA), all the tasks are offloaded to the network. Since central clouds are more energy efficient, BROA and FCOA begin to migrate the tasks from the edge clouds to the central clouds. That is why energy consumption decreases at $\gamma_d = 0.96$ in BROA ($\gamma_d = 0.98$ in FCOA).

The offloading problem formulated in our paper can be seen as a multi-objective optimization. Then, the goal of the proposed algorithms is to find solutions with high Pareto efficiency. Therefore, we give the achievable latency-energy regions in Figure 4(a) and (b) for the UE and the clouds, respectively. The points in Figure 4 are latency-energy pairs obtained with different γ_d. It could be seen that, BROA and FCOA have advantages compared with the others. When UE consumes 0.1 Joule energy per second, BROA and FCOA decrease the latency by 47.5% compared to ECO. Compared to Nash-overall, the latency reduction is about 18.8%when UE consumes 0.06 Joule per second. Moreover, from UE's perspective, the advantages of the proposed algorithm are more significant with smaller energy consumption. The reason is that less energy consumed on UE means that more tasks are offloaded to the clouds. Our proposed algorithms are more effective than the others on the utilization of cloud resources owing to the proposed marginal payoff function. Last but not least, we can see the curves of BROA and FCOA in Figure 4 almost coincide. It means that these two algorithms can achieve similar Pareto efficiency.

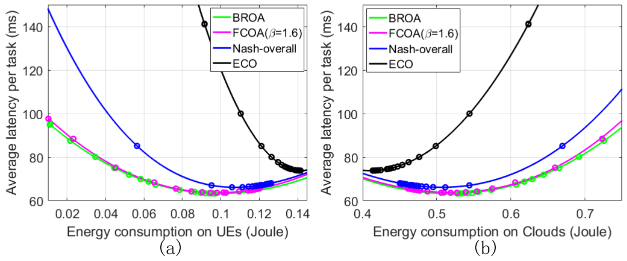


FIGURE 4. **Achievable latency-energy region with $|\mathcal{U}| = 60$, $|\mathcal{B}| = 2$, and $\gamma_e = 0.3$.**

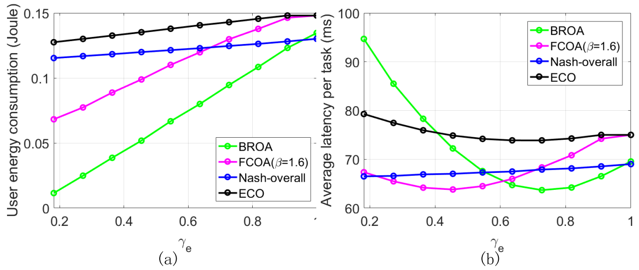


FIGURE 5. **Algorithm performance under different γ_e with $|\mathcal{U}| = 60$, $|\mathcal{B}| = 2$, and $\gamma_d = 0.9$.**

Now, we focus on the impacts of γ_e which indicates the relative importance of energy consumption on the UE. In Figure 5(a), the energy consumption on the UE monotonously increases with γ_e. The tendency implies that with a bigger γ_e, more tasks will be processed locally. The changes of latency curves in Figure 3(b) can be explained by the variations of offloading portion. We can see that as γ_e increases, the latency in BROA, FCOA, and ECO decreases firstly and then increases. The reason is that the processing latency per task on UE is a constant, i.e., 75ms. Meanwhile, the processing latency on edge clouds and central clouds are at least 25 ms and 9.3 ms, depending on their workload. Thus, the minimum task latency is achieved only when a proper portion of tasks is offloaded. In BROA (or FCOA), the minimum latency is achieved when γ_e equals 0.73 (or 0.45). We can also see that latency variations in ECO over γ_e is not as large as the cases in BROA and FCOA since ECO can only leverage the resources on the edge clouds. Nash-overall does not share a similar tendency with the other algorithms since its greedy revision rule cannot ensure to bring the performance improvement on the network cost.

To verify the effectiveness of the proposed algorithm under various circumstances, we conduct the simulations with different system settings, including K_e^{link} and κ.

Figure 6 shows the algorithm performance with different link capacities (K_e^{link}). Figure 6(a) demonstrates that cloud utilization increases with K_e^{link}. The reason is straightforward. The increase on K_e^{link} means that the cost motivated by transmission decreases. In this case, UE tends to offload more task which raises cloud utilization. Figure 6(b) plots the network cost improvement compared to LP. We can see that BROA and FCOA have advantages. Their network cost reduction is positively correlated with the cloud utilization. The correlation indicates the proposed marginal payoff function can properly schedule the resources on clouds. On the other hand, Nash-overall cannot efficiently use cloud resources. When K_e^{link} equals 0.1 Mbps, its network cost is even larger than the LP. Finally, the operators need to notice that when the link capacity exceeds 100 Mbps, additional bandwidth on backhaul links brings little offloading gains. It provides a decision basis for network planning.

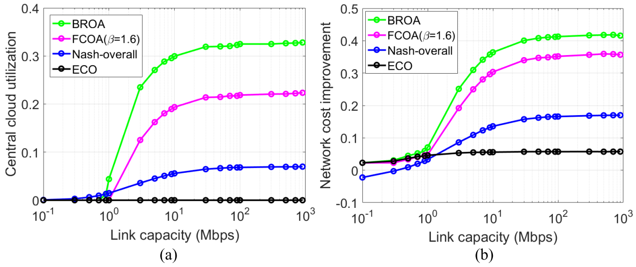


FIGURE 6. **Network performance with different link capacity on the backhaul and backbone networks. We set $|\mathcal{U}| = 60$, $|\mathcal{B}| = 2$, $\gamma_e = 0.3$, and $\gamma_d = 0.9$.**

Figure 7 demonstrates network cost variations affected by κ. Figure 7(a) shows that the offloading ratio curves of BROA, FCOA, and ECO share a similar tendency. Initially, the offloading ratio is zero. Then, it increases. After arriving at a maximum point, it monotonously decreases. To simplify our explanations, we measure the workload in terms of CPU cycles. The network cost motivated by processing a unit workload is a constant with respect to κ. Meanwhile, the cost for transmitting a unit workload is proportional to $1/\kappa$. When κ is small, the transmission cost is relatively high, which prevents the UE to offload the tasks. Thus, at this time, the offloading ratio is zero. As κ increases, the task offloading becomes more and more cost-efficient. In this period, the offloading ratio increases. When κ is larger, the clouds are overloaded, which reduces the offloading ratio. Figure 7(b) plots the network cost decreased due to utilizing the network resources. Apparently, BROA and FCOA outperform the others. Unlike the greedy rules in Nash-overall, the proposed marginal payoff function in these two algorithms can efficiently utilize the resources on the clouds to decrease the network cost. ECO is dominated since BROA and FCOA can use the resources in the backhaul and backbone networks.

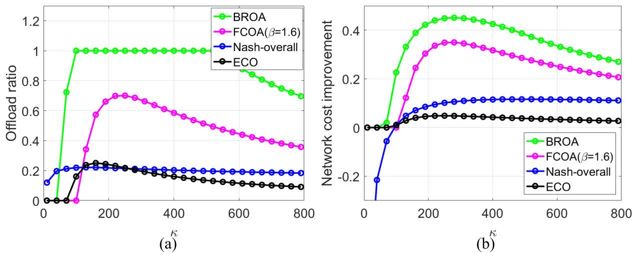


FIGURE 7. Network performance under κ with $|\mathcal{U}| = 60$, $|\mathcal{B}| = 2$, $\gamma_e = 0.3$, and $\gamma_d = 0.8$.

Table 5 and 6 emphasize the impacts of β. Table 5 compares the convergence speed of different algorithms in terms of re-routing times. Clearly, it shows that FCOA takes a significant advantage compared to BROA and Nash-overall. It can be seen that FCOA speeds up the convergence by at least 72.1% under 50 UE when $\beta = 1.2$. Also, the superiority of FCOA increases with β. For example, when β varies from 1.2 to 1.6, the advantage grows from 72.1% to 84.4%. Moreover, Table 5 indicates that the benefits introduced by β increases with the number of UE. FCOA can only reduce the re-routing times by 51.8% under 10 UE when β equals 1.2. However, the reduction on re-routing times increases to 71.6% when the number of UE equals 60.

TABLE 5. The overall re-routing times during the convergence.

No. UE	BROA	FCOA ($\beta = 1.2$)	FCOA ($\beta = 1.4$)	FCOA ($\beta = 1.6$)	Nash overall
10	168	81	53	38	174
20	364	152	110	78	437
30	565	179	134	103	595
40	652	203	152	116	697
50	867	241	189	135	914
60	1047	297	225	168	1089

Table 6 demonstrates the influences of β on accuracy. Among all the algorithms, BROA achieves the minimum network cost, which is consistent with our former conclusions. Nash-overall is even worse than LP sometimes owing to its selfish behaviors. Meanwhile, it is noticed that β leads to performance degradation. When β ranges from 1.2 to 1.6, the network cost improvement decreases 2.5% under 50 UE. Moreover, we can see that, in our simulations, the massive number of UE will weaken the advantages of BROA and FCOA. For example, BROA can reduce the network cost by 28.7% under 10 UE. The reduction shrinks to 17.3% under 60 UE. The reason is that since the network capacity is fixed, the portion of the migrated tasks decreases as workload increases (i.e., the number of UE increases). The network cost in the proposed algorithms will quarterly grow closer to the result of the LP.

TABLE 6. The network cost improvement compared to LP.

No. UE	BROA	FCOA ($\beta = 1.2$)	FCOA ($\beta = 1.4$)	FCOA ($\beta = 1.6$)	Nash overall
10	28.7%	27.1%	23.1%	20.8%	-1.0%
20	24.0%	22.1%	20.1%	18.8%	-2.0%
30	21.3%	19.7%	18.6%	16.3%	-2.4%
40	19.8%	18.6%	17.6%	15.5%	0.01%
50	18.7%	17.4%	16.4%	14.9%	0.8%
60	17.3%	15.8%	14.8%	13.5%	4.2%

VII. CONCLUSION

In this paper, we have investigated the computation offloading problem in the hierarchical MEC network. We generalize the assumption on the network layout and propose the topology-independent offloading algorithms which can balance the workload over the entire region of the network. The influences of resource-constraint links and devices in the backhaul and backbone network are emphasized by incorporating the task routing into offloading optimization. We convert the energy-latency aware offloading optimization into a routing problem so that the decision-making variables (including offloading decision, transmission power control, cloud selection, and task routing) are jointly optimized. Based on the game theory, a distributed energy-latency aware offloading algorithm, BROA, has been proposed. We show that a globally optimal solution can be achieved through UE collaboration. Furthermore, to make the proposed algorithm adapt to the time-varying environments, we propose a fast-converged algorithm FCOA. The performance of FCOA on convergence, accuracy, and time complexity is derived. The numerical results demonstrate that compared to traditional approaches, the proposed algorithms can significantly decrease the network cost in terms of energy consumption and latency.

In the future, we will focus on the migrations of our algorithms to the cases where multiple MEC applications coexist.

A more comprehensive discussion would be given on the wireless transmission. For example, inter-cell and intra-cell interference can be taken into consideration.

REFERENCES

[1] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, 4th Quart., 2017.

[2] T. X. Tran and D. Pompili, "Joint task offloading and resource allocation for multi-server mobile-edge computing networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 856–868, Jan. 2018.

[3] Y. Mao, J. Zhang, Z. Chen, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590–3605, Dec. 2016.

[4] F. Wang, J. Xu, X. Wang, and S. Cui, "Joint offloading and computing optimization in wireless powered mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 1784–1797, Mar. 2018.

[5] E. El Haber, T. M. Nguyen, and C. Assi, "Joint optimization of computational cost and devices energy for task offloading in multi-tier edge-clouds," *IEEE Trans. Commun.*, vol. 67, no. 5, pp. 3407–3421, May 2019.

[6] A. Kiani and N. Ansari, "Toward hierarchical mobile edge computing: An auction-based profit maximization approach," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 2082–2091, Dec. 2017.

[7] A. Kiani and N. Ansari, "Optimal code partitioning over time and hierarchical cloudlets," *IEEE Commun. Lett.*, vol. 22, no. 1, pp. 181–184, Jan. 2018.

[8] Q. Fan and N. Ansari, "Workload allocation in hierarchical cloudlet networks," *IEEE Commun. Lett.*, vol. 22, no. 4, pp. 820–823, Apr. 2018.

[9] P. Wang, C. Yao, Z. Zheng, G. Sun, and L. Song, "Joint task assignment, transmission, and computing resource allocation in multilayer mobile edge computing systems," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2872–2884, Apr. 2019.

[10] M. St-Hilaire, "Topological planning and design of UMTS mobile networks: A survey," *Wireless Commun. Mobile Comput.*, vol. 9, no. 7, pp. 948–958, Jul. 2009.

[11] R. Nadiv and T. Naveh, "Wireless backhaul topologies: Analyzing backhaul topology strategies," Ceragon, Tel Aviv-Yafo, Israel, White Paper, 2010, pp. 1–15.

[12] S. Orlowski, M. Pióro, A. Tomaszewski, and R. Wessäly, "SNDlib 1.0—Survivable network design library," *Networks*, vol. 55, no. 3, pp. 276–286, 2010.

[13] H. Guo and J. Liu, "Collaborative computation offloading for multiaccess edge computing over fiber–wireless networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4514–4526, May 2018.

[14] S.-R. Yang, Y.-J. Tseng, C.-C. Huang, and W.-C. Lin, "Multi-access edge computing enhanced video streaming: Proof-of-concept implementation and prediction/QoE models," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1888–1902, Feb. 2019.

[15] Y. Liu, J. Liu, A. Argyriou, and S. Ci, "MEC-assisted panoramic VR video streaming over millimeter wave mobile networks," *IEEE Trans. Multimedia*, vol. 21, no. 5, pp. 1302–1316, May 2019.

[16] J. Liu and Q. Zhang, "Code-partitioning offloading schemes in mobile edge computing for augmented reality," *IEEE Access*, vol. 7, pp. 11222–11236, 2019.

[17] D. Wang, B. Bai, K. Lei, W. Zhao, Y. Yang, and Z. Han, "Enhancing information security via physical layer approaches in heterogeneous IoT with multiple access mobile edge computing in smart city," *IEEE Access*, vol. 7, pp. 54508–54521, 2019.

[18] Z. Zhou, P. Liu, J. Feng, Y. Zhang, S. Mumtaz, and J. Rodriguez, "Computation resource allocation and task assignment optimization in vehicular fog computing: A contract-matching approach," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3113–3125, Apr. 2019.

[19] B. Blanco, J. O. Fajardo, I. Giannoulakis, E. Kafetzakis, S. Peng, J. Pérez-Romero, I. Trajkovska, P. S. Khodashenas, L. Goratti, M. Paolino, E. Sfakianakis, F. Liberal, and G. Xilouris, "Technology pillars in the architecture of future 5G mobile networks: NFV, MEC and SDN," *Comput. Standards Interfaces*, vol. 54, pp. 216–228, Nov. 2017.

[20] H. Guo, J. Zhang, and J. Liu, "FiWi-enhanced vehicular edge computing networks: Collaborative task offloading," *IEEE Veh. Technol. Mag.*, vol. 14, no. 1, pp. 45–53, Mar. 2019.

[21] V. Frascolla, F. Miatton, G. K. Tran, K. Takinami, A. De Domenico, E. C. Strinati, K. Koslowski, T. Haustein, K. Sakaguchi, S. Barbarossa, and S. Barberis, "5G-MiEdge: Design, standardization and deployment of 5G phase II technologies: MEC and mmWaves joint development for Tokyo 2020 Olympic games," in *Proc. IEEE Conf. Standards Commun. Netw. (CSCN)*, Sep. 2017, pp. 54–59.

[22] K. Kumar and Y.-H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" *Computer*, vol. 43, no. 4, pp. 51–56, Apr. 2010.

[23] S. Barbarossa, S. Sardellitti, and P. D. Lorenzo, "Communicating while computing: Distributed mobile cloud computing over 5G heterogeneous networks," *IEEE Signal Process. Mag.*, vol. 31, no. 6, pp. 45–55, Nov. 2014.

[24] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, "Energy-optimal mobile cloud computing under stochastic wireless channel," *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, pp. 4569–4581, Sep. 2013.

[25] I. Ketykó, L. Kecskés, C. Nemes, and L. Farkas, "Multi-user computation offloading as multiple knapsack problem for 5G mobile edge computing," in *Proc. Eur. Conf. Netw. Commun.*, Athens, Greece, Jun. 2016, pp. 225–229.

[26] C. Wang, F. R. Yu, C. Liang, Q. Chen, and L. Tang, "Joint computation offloading and interference management in wireless cellular networks with mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 66, no. 8, pp. 7432–7445, Aug. 2017.

[27] C. You, K. Huang, H. Chae, and B.-H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1397–1411, Mar. 2017.

[28] K. Zhang, Y. Mao, S. Leng, Q. Zhao, L. Li, X. Peng, L. Pan, S. Maharjan, and Y. Zhang, "Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks," *IEEE Access*, vol. 4, pp. 5896–5907, 2016.

[29] A. Sîrbu, C. Pop, C. Şerbănescu, and F. Pop, "Predicting provisioning and booting times in a Metal-as-a-service system," *Future Gener. Comput. Syst.*, vol. 72, pp. 180–192, Jul. 2017.

[30] N. Bessis, S. Sotiriadis, F. Pop, and V. Cristea, "Using a novel message-exchanging optimization (MEO) model to reduce energy consumption in distributed systems," *Simul. Model. Pract. Theory*, vol. 39, pp. 104–120, Dec. 2013.

[31] J. Xiong, H. Guo, and J. Liu, "Task offloading in UAV-aided edge computing: Bit allocation and trajectory optimization," *IEEE Commun. Lett.*, vol. 23, no. 3, pp. 538–541, Mar. 2019.

[32] Y. Jararweh, L. Tawalbeh, F. Ababneh, and F. Dosari, "Resource efficient mobile computing using cloudlet infrastructure," in *Proc. IEEE MSN*, Dec. 2013, pp. 373–377.

[33] A. Ceselli, M. Premoli, and S. Secci, "Mobile edge cloud network design optimization," *IEEE/ACM Trans. Netw.*, vol. 25, no. 3, pp. 1818–1831, Jun. 2017.

[34] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, fog: A survey and analysis of security threats and challenges," *Future Gener. Comput. Syst.*, vol. 78, pp. 680–698, Jan. 2018.

[35] I. Ahmad, T. Kumar, M. Liyanage, J. Okwuibe, M. Ylianttila, and A. Gurtov, "Overview of 5G security challenges and solutions," *IEEE Commun. Standards Mag.*, vol. 2, no. 1, pp. 36–43, Mar. 2018.

[36] L. Xiao, X. Wan, C. Dai, X. Du, X. Chen, and M. Guizani, "Security in mobile edge caching with reinforcement learning," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 116–122, Jun. 2018.

[37] L. Xiao, Y. Li, G. Han, G. Liu, and W. Zhuang, "PHY-layer spoofing detection with reinforcement learning in wireless networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 12, pp. 10037–10047, Dec. 2016.

[38] K. Zhang, Y. Mao, S. Leng, Y. He, and Y. Zhang, "Mobile-edge computing for vehicular networks: A promising network paradigm with predictive offloading," *IEEE Veh. Technol. Mag.*, vol. 12, no. 2, pp. 36–44, Jun. 2017.

[39] H. Guo, J. Liu, and J. Zhang, "Efficient computation offloading for multi-access edge computing in 5G HetNets," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.

[40] A. Sfrent and F. Pop, "Asymptotic scheduling for many task computing in big data platforms," *Inf. Sci.*, vol. 319, pp. 71–91, Oct. 2015.

[41] J. Wang, J. Hu, G. Min, W. Zhan, Q. Ni, and N. Georgalas, "Computation offloading in multi-access edge computing using a deep sequential model based on reinforcement learning," *IEEE Commun. Mag.*, vol. 57, no. 5, pp. 64–69, May 2019.

[42] M.-A. Vasile, F. Pop, R.-I. Tutueanu, V. Cristea, and J. Kołodziej, "Resource-aware hybrid scheduling algorithm in heterogeneous distributed computing," *Future Gener. Comput. Syst.*, vol. 51, pp. 61–71, Oct. 2015.

[43] H. Guo, J. Liu, J. Zhang, W. Sun, and N. Kato, "Mobile-edge computation offloading for ultradense IoT networks," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4977–4988, Dec. 2018.

[44] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic Game Theory*. Cambridge, U.K.: Cambridge Univ. Press, 2007.

[45] D. Fotakis, S. Kontogiannis, and P. Spirakis, "Selfish unsplittable flows," *Theor. Comput. Sci.*, vol. 348, pp. 226–239, Dec. 2005.

[46] H. Borowski and J. R. Marden, "Fast convergence in semianonymous potential games," *IEEE Trans. Control Netw. Syst.*, vol. 4, no. 2, pp. 246–258, Jun. 2017.

[47] M. Gao, B. Addis, M. Bouet, and S. Secci, "Optimal orchestration of virtual network functions," *Comput. Netw.*, vol. 142, pp. 108–127, Sep. 2018.

[48] S. Ko, K. Han, and K. Huang, "Wireless networks for mobile edge computing: Spatial modeling and latency analysis," *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5225–5240, Aug. 2018.

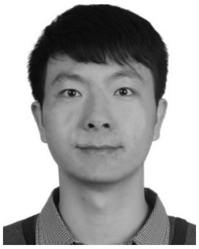
[49] W. H. Sandholm and C. Ansell, *Population Games and Evolutionary Dynamics*. Cambridge, MA, USA: MIT Press, 2010.

[50] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou, "The complexity of computing a Nash equilibrium," *SIAM J. Comput.*, vol. 39, no. 1, pp. 195–259, 2009.

[51] X. Chu, D. Lopez-Perez, Y. Yang, and F. Gunnarsson, *Heterogeneous Cellular Networks: Theory, Simulation and Deployment*. Cambridge, U.K.: Cambridge Univ. Press, 2013.

[52] Y. Ge, Y. Zhang, Q. Qiu, and Y.-H. Lu, "A game theoretic resource allocation for overall energy minimization in mobile cloud computing system," in *Proc. ACM/IEEE Int. Symp. Low Power Electron. Design*, Aug. 2012, pp. 279–284.

BINWEI WU received the B.E. degree in communication engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2013, where he is currently pursuing the Ph.D. degree in information and communication engineering with the National Key Laboratory of Science and Technology on Communications. His research interests include novel network architecture, MEC networks, and vehicular networks.



JIE ZENG (M'09–SM'16) received the B.S. and M.S. degrees in electronic engineering from Tsinghua University, in 2006 and 2009, respectively. He has published three books and over 100 journal and conference papers. He holds more than 30 Chinese and seven international patents. His research interests include novel network architecture, ultra-dense networks, and novel multiple access. He received the science and technology award of Beijing, in 2015, and the best cooperation award of Samsung Electronics, in 2016.

LU GE received the B.S. degrees in electrical and systems engineering from the Chongqing University of Posts and Telecommunications, China, in 2006, and the M.S. and Ph.D. degrees in electrical and systems engineering from Loughborough University, Leicestershire, U.K., in 2007 and 2011, respectively. She is currently a Postdoctoral Researcher with Tsinghua University, China. Her research interests include novel network architecture, ultra-reliable and low latency communication, and novel multiple access.

XIN SU (M'03–SM'15) received the M.S. and Ph.D. degrees in electronic engineering from the University of Electronic Science and Technology of China (UESTC), in 1996 and 1999, respectively. He is currently a Full Professor with the Research Institute of Information Technology, Tsinghua University. He is also the Chairman of the IMT-2020 (5G) Wireless Technology Work Group in Ministry of Industry and Information Technology (MIIT) of People's Republic of China and the Vice Chairman of the Innovative Wireless Technology Work Group of China Communications Standards Association (CCSA). He has published over 100 articles in the core journals and important conferences, and he holds more than 30 patents. His research interests include broadband wireless access, wireless and mobile network architecture, self-organizing networks, software defined radio, and cooperative communications.

YOUXI TANG was born in Xinyang, Henan, China, in 1964. He received the B.E. degree in radar engineering from the College of PLA Ordnance, Shijiazhuang, China, in 1985, and the M.S. and Ph.D. degrees in communications and information systems from the University of Electronic Science and Technology of China, Chengdu, China, in 1993 and 1997, respectively, where he has been a Professor with the National Key Laboratory of Science and Technology on Communications, since 2000. From 1998 to 2000, he was a Program Manager with Huawei Technologies Company Ltd., Shanghai, China, where he was involved in the area of IS-95 mobile communications and third-generation mobile communications. His general research interests include spread spectrum systems and wireless mobile systems with emphasis on signal processing in communications.