A Distributed Computation Offloading Strategy in Small-Cell Networks Integrated With Mobile Edge Computing

Lichao Yang, Heli Zhang, Xi Li, Hong Ji, *Senior Member, IEEE*, and Victor C. M. Leung, *Fellow, IEEE*

***Abstract*— Mobile edge computing is conceived as an appealing technology to enhance cloud computing capability of mobile devices (MDs) at the edge of the networks. Although some researchers use the technology to address the intensive tasks' high computation needs of MDs in small-cell networks (SCNs), most of them ignore considering the interests interaction between small cells and MDs. In this paper, we study a distributed computation offloading strategy for a multi-device and multi-server system based on orthogonal frequency-division multiple access in SCNs. First, to satisfy the interest requirements of different MDs and analyze the interactions among multiple small cells, we formulate a distributed overhead minimization problem, aiming at jointly optimizing energy consumption and latency of each MD. Second, to ensure the individuals of different MDs, we formulate the proposed overhead minimization problem as a strategy game. Then, we prove the strategy game is a potential game by the feat of potential game theory. Moreover, the potential game-based offloading algorithm is proposed to reach a Nash equilibrium. In addition, to guarantee the performance of the designed algorithm, we consider the lower bound of iteration times to derive the worst case performance guarantee. Finally, the simulation results corroborate that the proposed algorithm can effectively minimize the overhead of each MD compared with different other existing algorithms.**

***Index Terms*— Mobile edge computing, small cell networks, computation offloading, Nash equilibrium, potential game.**

I. Introduction

DRIVEN by the exploding quantity of mobile devices (MDs), the scale of abundant applications becomes increasing large [1], such as natural language processing, virtual reality and highly-interactive online gaming to facial recognition, etc. However, due to the limitation of battery power, computation capacity and physical size of MDs, those computation-intensive applications can not be performed smoothly on the MDs [2], [3]. To tackle this issue, in recent years, the emergence of mobile edge computing provides a beam of light to the conflict between resource-limited MDs and computation-intensive applications.

In mobile edge computing (MEC) paradigm, MEC can enhance computation and storage capabilities at the edge of small cell networks (SCNs) by deploying MEC servers [4], [5]. In addition, MEC also significantly reduces latency, provides high bandwidth and prolongs the battery lifetime of MDs, which is widely used as an effective way to liberate the MDs from heavy computation workloads. Thus, a significant amount of researches focus on computation offloading in 5G communication system. Moreover, consider that SCN [6]–[8] is an important network architecture in future 5G, thus, integrating MEC with SCN has attracted growing researchers' interests in academia and industry.

With MEC as the enabling technology, offloading computation-intensive tasks via two approaches has been studied, centralized computation offloading model and distributed computation offloading model respectively. For the centralized computation offloading model [9]–[12], where heavy offloading information is collected in the dedicated network management nodes in real-time. This may be difficult to implement when MEC servers are widely deployed in practice. It will lead the system to hiccup if the controller is attacked or in a hardware failure. Therefore, a distributed computation offloading model is desired [13], [14]. Specially, for the distributed computation offloading model, in which each MEC server can choose MDs by self-selection according to the information what they collect. In this case, if one server is damaged, the whole network can still work normally. Simultaneously, the distributed computation offloading model is flexible and scalable. Above all, we will take the distributed computation offloading model to solve our proposed problem in this paper. Unfortunately, the distributed computation offloading model will impose huge overhead when switched, so it is of great necessary to improve overhead of the system.

In this paper, we study the computation offloading in 5G MEC, featured by its SCN architecture. Specially, we focus on the design of distributed and efficient computation offloading

Manuscript received January 4, 2018; revised July 18, 2018 and October 7, 2018; accepted October 8, 2018; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor J. Huang. This work was supported in part by the National Natural Science Foundation of China under Grant 61671088, in part by the National Natural Science Foundation of China under Grant 61771070, in part by the National Science and Technology Major Project of the Ministry of Science and Technology of China under Grant 2017ZX03001014, and in part by the BUPT Excellent Ph.D. Students Foundation under Grant CX2018203. *(Corresponding author: Heli Zhang.)*

L. Yang, H. Zhang, X. Li, and H. Ji are with the Key Laboratory of Universal Wireless Communications, Ministry of Education, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: lichaoyang@bupt.edu.cn; zhangheli@bupt.edu.cn; lixi@bupt.edu.cn; jihong@bupt.edu.cn).

V. C. M. Leung is with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC V6T 1Z4, Canada (e-mail: vleung@ece.ubc.ca).

Digital Object Identifier 10.1109/TNET.2018.2876941

strategy. First, an overhead minimization problem is formulated. Then an efficient potential game based offloading algorithm (PGOA) [15] is proposed, taking the interests of different MEC-MBS and MDs into account. The algorithm is proved to have a threshold-based structure offloading decision function, which yields the tasks to be executed as a whole either locally at the MD or remotely at the MEC server. Finally, simulation results verify the theoretical analysis and the performance of the PGOA, which not only significant improves the energy consumption of the system, but also reduces the latency of the system. Furthermore, our contributions can be summarized as follows:

- *Study the SCN Integrated With MEC:* Compared with the conventional network architecture, SCN hits a perfect match with MEC technology. The characteristic of integrating MEC and SCN poses a new challenge to effectively allocate communication and computation resources. To address the challenge of server load balancing, we formulate an overhead minimization problem under the communication and computation resources constraints.
- *Formulate the Proposed Problem as a Potential Game Among Multi-Device and Multi-MEC:* In the paper, we study a multi-device and multi-MEC scenario where we consider the interest requirements of different MDs and the interactions among multiple small cells. Thus, the potential game becomes a powerful tool for distributed computation offloading approach to solve overhead minimization problem among multi-device and multi-MEC.
- *Construct a Threshold-Based Structure Offloading Decision Function:* Since we adopt a binary computation offloading model in 5G SCN integrated with MEC, and the performance of which is susceptible to suffer from the offloading latency, energy consumption and transmission interference effect, we construct a threshold-based structure offloading decision function to judge the tasks of MDs executed locally or remotely.
- *Design an Efficient PGOA:* We design PGOA to solve the distributed computation offloading problem, and derive a Nash Equilibrium (NE).

The rest of this paper is organized as follows. We first discuss the related work in Section II. We next introduce the system model in Section III. We then model the overhead minimization problem in terms of the game theory in Section IV. We design an efficient PGOA to achieve an NE in Section V. To measure the performance of the proposed algorithm, we analyze the convergence of PGOA and introduce the efficiency ratio (ER) over the sub-optimal offloading decision in Section VI. What's more, the simulation results are presented in section VII. Finally, we conclude this paper in section VIII.

II. Related Work

We survey existing MEC concepts proposed in the literatures integrating the MEC functionalities into the mobile networks. Recent years have witnessed encouraging process on this topic for both multi-user and single-server as well as multi-user and multi-server MEC systems. For a multi-user and single-server [16]–[19] MEC system, You *et al.* [16] studied a sum mobile energy consumption minimization problem for multi-device and single edge cloud server. Sardellitti *et al.* [17] considered an MIMO multi-cell system where multiple mobile users ask for computation offloading to a common cloud server with aiming at minimizing the overall users' energy consumption. Later in [18], a game theoretic approach for computation offloading to optimize the utility of the service providers while also reducing the energy cost and the tasks execution time at smart devices was proposed. Another overhead minimization problem of the MDs in [19] was addressed, and a distributed computation offloading for multi-user MEC at a single cloud was designed using game theory with the infinite computation capacity of the edge cloud. With the increasing of the number of intensive-computation tasks and the complexity of SCN, the single server supply is one major limitation for powerful computational resources at the edges, so the multi-user and multi-server [20]–[22] MEC system has attracted significant attention in recent years. Zhao *et al.* [20] deployed multiple local cloud servers at the edge of base stations for offloading, optimizing the offloading delay within the limited computational resource of the local cloud servers. In addition, Yang *et al.* [21] studied the energy consumption for offloading in multi-user and multi-server scenario where the transmission scheduling was carried over both the fronthaul and backhaul links. The cooperation among multiple MDs in [22] was presented to realize the task scheduling by constructing a potential game, aiming at minimizing the overhead of each MD. In view of prior works, most of them assumed that the computation capacity of the edge cloud is infinite. Along a different line, in this paper, we consider a multi-user and multi-server scenario where we minimize the overhead of each MD in cooperative MEC by joint communication and computation resources allocation with finite computation capacity in SCN.

The approach of computation offloading strategies is now emerging as a key factor affecting network performance. However, most of scholars attached importance to the research on the centralized computation offloading strategy. For example, Chen *et al.* [19] combined energy and time together as the network cost, depended on which, users could determine whether the computation task should be offloaded. Zhang *et al.* [23], [24] designed an energy-efficient computation offloading scheme with the aim of minimizing the energy consumption.

More recently, some scholars also did works on the distributed computation offloading strategy in Internet of Things (IoT) or virtual machines (VMs). For instance, Sun *et al.* [25] studied the cost-effective wireless services for high-speed railway to do distributed and dynamic resource management. Furthermore, Xie *et al.* [26] designed a multi-dimensional pricing mechanism based on two-side market game, and proposed a distributed price-adjustment algorithm for resource allocation and offloading scheduling. Different from those computation offloading studies in 5G SCN integrated with MEC, in this article, we consider a multi-device and multi-MEC scenario where each MD adopts binary offloading model which involves both

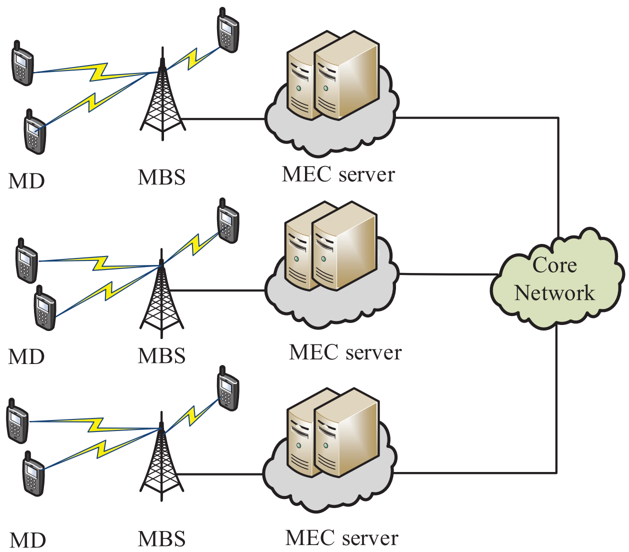

Fig. 1. An illustration of small cell network integrated with MEC.

communication and computation resources allocation, aiming at minimizing the overhead of each mobile device. Moreover, by leveraging the potential game, we design an efficient and distributed potential game based offloading algorithm.

III. System Model

In this section, the system model adopted in this work is described. We first describe SCN integrated with MEC model, then, we present communication model and computation model in details.

A. Small Cell Network Integrated With MEC Model

In this paper, we consider a SCN integrated with MEC system model, which is composed of multiple devices and multiple Micro base stations (MBSs), as shown in Fig.1. In particular, the MBSs, defined by $\mathbb{N} = \{1, \ldots, N\}$, are connected to the core network through wired optical fibers. As mentioned before, MBSs are installed with MEC servers, with which, MBSs are endowed with MEC function and able to provide the enhanced computation capabilities at the edge of SCN. We call MBSs with MEC servers as MEC-MBSs. Let $\mathbb{I} = \{1, \ldots, I\}$ denote the set of MDs, and each MD is associated with only one MEC server via the MBS. For the channel access scheme, we adopt a multi-device and multi-MEC orthogonal frequency-division multiple access (OFDMA) scheme where the spectrum in each MEC-MBS is divided into $\mathbb{M} = \{1, 2, \ldots, M\}$ channels, and channels are orthogonal in one MEC-MBS. Since we aim at achieving distributed and efficient task offloading scheduling for all MDsk, an overhead minimization problem is formulated, not only considering the correlations and interests between MDs and MEC-MBSs, but also taking multiple resource constraints into account. Considering that uncertainty of channels and competition of the limited computation resource, the overhead minimization problem for offloading will jointly optimizes the communication and computation resources allocation from communication model and computation model.

B. Communication Model

Communication model for wireless access is presented as follows. Denote $a_{i,n}^m \in \{0, 1\}$ as the decision set of channels and MEC servers of MD i. Especially, $a_{i,n}^m = 0$ denotes that MD i chooses not to access MEC-MBS n via channel m. $a_{i,n}^m = 1$ denotes MD i access to MEC-MBS n through channel m. Denote $a = \{a_1, a_2, \ldots, a_i, \ldots, a_I\}$ as the computing mode decision profile of all the MDs, where $a_i \in \{0, 1\}$ is the computing mode selection of MD i. Specially, $a_i = \sum\limits_n \sum\limits_m a_{i,n}^m = 1$ means MD i chooses to execute its computation task remotely on the MEC-MBS n via channel m. We have $a_i = \sum\limits_n \sum\limits_m a_{i,n}^m = 0$, if MD i decides to compute its task locally at the MD. Assume MD i accesses MEC-MBS n via channel m, and we can obtain the data transmission rate as:

$$r_{i,n}^m = W \log_2(1 + \frac{p_{i,n} g_{i,n}^m}{\sigma^2 + K_{i,n}^m}) \tag{1}$$

where W is channel bandwidth, $p_{i,n}$ is MD i's transmission power. $g_{i,n}^m$ means channel gain for the link between MD i and MEC-MBS n via channel m. $K_{i,n}^m = \sum\limits_{l=1, l \neq n}^{N} \sum\limits_{j=1, j \neq i}^{I} a_{j,l}^m p_{j,l} g_{j,n}^m$ denotes the interference of channel m suffered from other MDs which choose the same channel when offloaded. Since we adopt OFDMA channel access scheme, only the interference between MBSs is considered. σ^2 is the Gaussian noise.

C. Computation Model

We then introduce the computation model. In this article, we just study the binary offloading problem which requires a task to be executed as a whole either locally at the MD or remotely at the MEC-MBS [27], calling them as the local computing and the edge computing respectively. Specially, we introduce the computation model from the next two computing modes: local computing and edge computing, respectively. In addition, assume that MD i has a computation task $L_i = \{b_i, w_i, t_i^{max}\}$ to be executed, where denote b_i as the size of computation task of MD i, and w_i represents the total computation capability (i.e., CPU cycles) to complete the computation task. In our article, we set η_i denoting CPU cycles required for each MB of the task, which is a fixed constant for each MD i, i.e., $\eta_i = w_i / b_i$. t_i^{max} is the maximum latency that can be tolerated by MD i.

1) Local Computing: When MD i selects local computing, the overhead of the local computing contains two parts: the computation task's local execution latency t_i^l and energy consumption E_i^l. Especially, we have $t_i^l = w_i / f_i^l$ and $E_i^l = w_i \varepsilon^l$, where f_i^l denotes the computation capability (i.e., CPU cycles per second) of MD i, which is different for each MD according to the cloud computing service subscribed from the telecom operator, and ε_i^l is the energy consumption coefficient for per CPU cycle of MD i. To guarantee both energy and latency minimization at MD, we do a weighting sum of energy and latency, named them as overhead [19], [28]. We first give the following local overhead as

$$Z_i^l = \lambda_1 t_i^l + \lambda_2 E_i^l \tag{2}$$

where $0 \leq \lambda_1, \lambda_2 \leq 1$ ($\lambda_1 + \lambda_2 = 1$) denotes the weighting parameters of execution latency and energy consumption for MD i respectively. In order to reflect the characteristic of different research scenario, we set different weighting parameters which can reflect general demands of various research scenario. For example, if we study the overhead of updating website content, the survey finds that latency is the main effect factor for loss in website performance. Thus, we can put larger weight on latency for the overhead minimization problem. If the research scenario significantly focuses on energy sensitive applications, we will put larger weight on energy consumption. In this paper, since we mainly study the overhead of the lightweight, flexible and small electronic MDs, we set the weight of parameter λ_1 equal to that of parameter λ_2, and weights of the parameters λ_1 and λ_2 remain unchanged for each offloading process.

2) Edge Computing: When MD i chooses to execute its computation task on MEC-MBS n via channel m, the overhead of the edge computing mainly comes from the computation task transmission process and the computation task computation process. First, we compute the total latency of the edge computing, which is composed of latency to computation task transmission $t_{i,n}^{m,s} = b_i/r_{i,n}^m$ and latency to the computation task computation on MEC-MBS n, $t_{i,n}^x = w_i/f_{i,n}^x$ respectively, where $f_{i,n}^x$ denotes the computation capability (i.e., CPU cycles per second) of MEC-MBS n allocated to MD i. Accordingly, the total latency of MD i in the edge computing is achieved by

$$t_{i,n}^{m,e} = t_{i,n}^{m,s} + t_{i,n}^x \tag{3}$$

Secondly, we consider the energy consumption from MD i to MEC-MBS n via the channel m in the edge computing, denoted by $E_{i,n}^{m,e}$, which is mainly composed of transmission energy consumption $E_{i,n}^{m,s} = p_{i,n}b_i/r_{i,n}^m$ and execution energy consumption $E_{i,n}^x = w_i\varepsilon_n$, where ε_n is the energy cost of each MEC server for implementing a CPU cycle. Thus, the total energy consumption of MD i in the edge computing is

$$E_{i,n}^{m,e} = E_{i,n}^{m,s} + E_{i,n}^x \tag{4}$$

Above all, we can compute the total overhead in terms of latency and energy consumption as

$$Z_i^e = \lambda_1 \sum_n \sum_m a_{i,n}^m t_{i,n}^{m,e} + \lambda_2 \sum_n \sum_m a_{i,n}^m E_{i,n}^{m,e} \tag{5}$$

In this article, because the size of computation outcome is smaller than that of the computation input in general, we neglect the down-link latency when MEC-MBSs send the computation results back to the MDs [5], [19].

IV. Problem Formulation

With the system model described above, we next formulate an overhead minimization problem.

As communication model defined, notice that a_i denotes the computing mode selection of MD i, and a_{-i} denotes the computing mode selection by all other MDs except MD i. $\{Z_i\}_{i\in\mathbb{I}}$ denotes the overhead function of each MD i. Then the overhead minimization problem is formulated as

$$\min_{a_i\in\{0,1\}} Z_i(a_i, a_{-i}), \quad \forall i \in \mathbb{I} \tag{6}$$

Particularly, the objective function Z_i is given by

$$Z_i(a_i, a_{-i}) = \begin{cases} Z_i^l(1-a), & if\ a_i = 0. \\ Z_i^e(a), & if\ a_i = 1. \end{cases} \tag{7}$$

Considering those MDs adopt a binary computation offloading rule, where MD i needs to determine whether to be processed locally at its device or executed remotely at the MEC-MBS, the computing mode selection will be susceptible to suffer from the offloading latency, energy consumption and transmission interference effect. Thus, to decrease the impact of improper computing mode selection on the quality of service (QoS) of MDs, we construct a threshold-based structure offloading decision function which jointly takes the load constraint, latency constraint, interference constraint and energy consumption constraint into account to judge the tasks executed locally or remotely. We elaborate on those constraints as follows.

- Load constraint. As the computation resource of each MEC-MBS is limited, the load constraint prevents executing too many tasks on the same MEC-MBS. Concurrently, let c_n denote the computation resource of MEC-MBS n, all total computation resource allocated to computation tasks executed on MEC-MBS should not surpass the computation resource c_n of MEC-MBS n, which can be presented by: $\sum_{i\in\mathbb{I}} \sum_{m\in\mathbb{M}} a_{i,n}^m f_{i,n}^x \leq c_n, n \in \mathbb{N}$.
- Latency constraint. We set a threshold parameter κ_1 to judge when MD i can execute computation-intensive tasks by the edge computing. When MD i chooses the edge computing, the decision condition can be given by: $t_{i,n}^{m,e} < \kappa_1 t_i^l$ $(0 < \kappa_1 \leq 1)$.
- Interference constraint. When MD i accomplishes those computation-intensive tasks by the edge computing, in order to ensure the quality of received signals through the MEC-MBS, the total interference noise power that MD measures, which should be less than the signal power of MD i. Let κ_2 denote the threshold, which can be expressed by: $(\sigma^2 + \sum_{l=1,l\neq n}^{N} \sum_{j=1,j\neq i}^{I} a_{j,l}^m p_{j,l} g_{j,n}^m)/p_{i,n} g_{i,n}^m < \kappa_2 (0 < \kappa_2 \leq 1)$.
- Energy consumption constraint. If MD i selects edge computing, the energy consumption of the edge computing should be not higher than that of the local computing. In order to meet user-specific demands, we set κ_3 as the weighting parameter of locally energy consumption, which can be presented by: $E_{i,n}^{m,e} < \kappa_3 E_i^l$ $(0 < \kappa_3 \leq 1)$.

In order to solve this overhead minimization problem, we need to resort to the potential game theory. In terms of the finite improvements and NE property of the potential game, we design an offloading algorithm to obtain a sub-optimal solution.

V. Potential Game Based Offloading Scheme

We next propose an efficient potential game based offloading scheme among multi-device and multi-MEC. Particularly, we first formulate the overhead minimization problem as a strategy game, then we introduce a potential function to prove the formulated game is a potential game. Then, we analyze the structural property of the potential game and show that the potential game admits an NE within finite improvements. Based on the characteristic of potential game, we design a potential game based offloading algorithm (PGOA).

A. Potential Game Formulation

In this section, we introduce a potential game to prove the strategy game related to overhead minimization problem is a potential game in fact. First, we formulate the overhead minimization problem above as a strategy game model $\Gamma = (\mathbb{I}, \{A_i\}_{i\in\mathbb{I}}, \{Z_i\}_{i\in\mathbb{I}})$, where we regard each MD as one player, so $\mathbb{I}$ denotes the set of all the players, $\{A_i\}_{i\in\mathbb{I}} \in \{a_1, a_2, \ldots, a_I\}$ denotes the strategies set (i.e., computing mode selection set) of each player i. Next, we will elaborate on that the strategy game is a potential game.

Definition 1: A game is called a potential game with the potential function, if the potential function P satisfies: $P : a \to R$, for every $i \in \mathbb{I}, a_{-i} \in \Pi_{n\neq i} A_n$, and $a_i', a_i \in A_i$, if

$$Z_i(a_i', a_{-i}) < Z_i(a_i, a_{-i}), \tag{8}$$

we have

$$P_i(a_i', a_{-i}) < P_i(a_i, a_{-i}). \tag{9}$$

We will give the potential function P in the followings.

Definition 2: Define $C = \beta_1\kappa_1+\beta_2\kappa_2+\beta_3\kappa_3$ as a threshold value of computation offloading. If MD i obtains one threshold function value $C_{i,n}^m$ lower than the threshold value C, the MD will choose local computing, otherwise, edge computing. Name those MDs which choose edge computing as the qualified MDs. The followings are the threshold function:

$$C_{i,n}^m = \beta_1(\frac{t_{i,n}^{m,e}}{t_i^l}) + \beta_2(\frac{\sigma^2 + \sum_{l=1,l\neq n}^{N}\sum_{j=1,j\neq i}^{K} a_{j,l}^m p_{j,l} g_{j,n}^m}{p_{i,n} g_{i,n}^m}) + \beta_3(\frac{E_{i,n}^{m,e}}{E_i^l}) \tag{10}$$

where $\beta_j (j = \{1, 2, 3\})$ is the weighting parameters of latency, interference and energy consumption respectively, $0 \leq \beta_j \leq 1$ and $\sum_{j=1}^{3}\beta_j = 1$. Since the total overhead of the system is affected by those three parameters, we define the threshold function which jointly optimizes latency, interference and energy consumption. In (17), the first item states the effect of the latency. When the latency cost of edge computing is low enough, the MD will select the edge computing, not the local computing. The second item indicates the effect of interference. If one MD chooses the edge computing, the SINR should be higher than the threshold of the SINR to ensure computation task transmission normally. The third item means the effect of the energy consumption. If the energy consumption of computing tasks via the edge computing is lower than the local computing, MD i will offload the computation tasks to the MEC-MBS via the edge computing.

According to the above analysis, we give the following threshold strategy:

$$a_{i,n}^m = \begin{cases} 1, & if\ C_{i,n}^m < C \\ 0, & if\ C_{i,n}^m > C. \end{cases} \tag{11}$$

We can define the potential function as

$$P(a) = \frac{1}{2}\sum_{n=1}^{N}\sum_{i=1}^{I}\sum_{l\neq n}\sum_{j\neq i} C_{i,n}^m C_{j,l}^m I_{\{a_i=a_j:m\}} I_{\{a_i=1\}} + \sum_{n=1}^{N}\sum_{i=1}^{I} C_{i,n}^m C I_{\{a_i=0\}} \tag{12}$$

The indicator function $I_{\{A\}}$, which shows the true or false of the event A. $I_{\{A\}} = 1$ shows the event A is true, otherwise, $I_{\{A\}} = 0$. Thereinto, the event A means that the MD i and MD j choose the same channel m during offloading process.

Next, we will we will prove the strategy game is a potential game. First, we formulate the overhead minimization problem as a strategy game. Suppose that MD i wants to update its current decision a_i, there just right exists a decision a_i', which satisfies $Z_i(a_i', a_{-i}) < Z_i(a_i, a_{-i})$. Since the latency and energy of the overhead and the interference are monotonously increasing in terms of variable of the overhead function, MD i will update its current decision a_i to a_i', and the formula $\sum_{l=1}\sum_{j=1} C_{j,l}^m I_{\{a_j\neq a_i':m\}} < \sum_{l=1}\sum_{j=1} C_{j,l}^m I_{\{a_j\neq a_i:m\}}$ is satisfied. According to the definition of potential game, we will prove the strategy game is a potential game from the following three cases:

1) Suppose the current MD's decision is $a_i = 0$, if MD updates its current decision $a_i = 0$ to $a_i' = 1$. Based on $Z_i(a_i', a_{-i}) < Z_i(a_i, a_{-i})$, we know $\sum_{l\neq n}\sum_{j\neq i} C_{j,l}^m I_{\{a_j=a_i':m\}} < C$ we know that

$$\begin{aligned} P_i(a_i', a_{-i}) - P_i(a_i, a_{-i}) &= \frac{1}{2}C_{i,n}^m \sum_{l\neq n}\sum_{j\neq i} C_{j,l}^m I_{\{a_j=a_i':m\}} \\ &\quad + \frac{1}{2}\sum_{l\neq n}\sum_{j\neq i} C_{i,n}^m C_{j,l}^m I_{\{a_i'=a_j:m\}} \\ &\quad - C_{i,n}^m C \\ &= C_{i,n}^m \sum_{l\neq n}\sum_{j\neq i} C_{j,l}^m I_{\{a_j=a_i':m\}} \\ &\quad - C_{i,n}^m C < 0 \end{aligned} \tag{13}$$

2) Suppose the current MD's decision is $a_i = 1$, if the MD updates its current decision $a_i = 1$ to $a_i' = 0$, we know $C < \sum_{l\neq n}\sum_{j\neq i} C_{j,l}^m I_{\{a_j=a_i':m\}}$, and get that

$$\begin{aligned} P_i(a_i', a_{-i}) - P_i(a_i, a_{-i}) &= C_{i,n}^m C \\ &\quad - \frac{1}{2}C_{i,n}^m \sum_{l\neq n}\sum_{j\neq i} C_{j,l}^m I_{\{a_j=a_i:m\}} \\ &\quad - \frac{1}{2}\sum_{l\neq n}\sum_{j\neq i} C_{i,n}^m C_{j,l}^m I_{\{a_i=a_j:m\}} \\ &= C_{i,n}^m C \\ &\quad - C_{i,n}^m \sum_{l\neq n}\sum_{j\neq i} C_{j,l}^m I_{\{a_j=a_i:m\}} < 0 \end{aligned} \tag{14}$$

3) Suppose the current MD's decision is $a_i = 1$, if the MD updates its current decision $a_i = 1$ to $a_i' = 1$. According to

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

the known condition, we know that

$$\begin{aligned} P_i(a_i', a_{-i}) - P_i(a_i, a_{-i}) &= \frac{1}{2} C_{i,n}^m \sum_{l \neq n} \sum_{j \neq i} C_{j,l}^m I_{\{a_j = a_i' : m\}} \\ &\quad + \frac{1}{2} \sum_{l \neq n} \sum_{j \neq i} C_{i,n}^m C_{j,l}^m I_{\{a_i' = a_j : m\}} \\ &\quad - \frac{1}{2} C_{i,n}^m \sum_{l \neq n} \sum_{j \neq i} C_{j,l}^m I_{\{a_j = a_i : m\}} \\ &\quad - \frac{1}{2} \sum_{l \neq n} \sum_{j \neq i} C_{i,n}^m C_{j,l}^m I_{\{a_i = a_j : m\}} \\ &= C_{i,n}^m \sum_{l \neq n} \sum_{j \neq i} C_{j,l}^m I_{\{a_j = a_i' : m\}} \\ &\quad - C_{i,n}^m \sum_{l \neq n} \sum_{j \neq i} C_{j,l}^m I_{\{a_j = a_i : m\}} < 0 \end{aligned} \tag{15}$$

Based on the analysis what we have obtained, we can see that the reduction in its overhead function will lead to the reduction in the potential function. As a result, the formulated strategy game is a potential game by constructing the potential function P successfully.

B. Existence of the Nash Equilibrium

Definition 3: The game $\Gamma = (\mathbb{I}, \{A_i\}_{i \in \mathbb{I}}, \{Z_i\}_{i \in \mathbb{I}})$, each player's strategy a_i^* is the best strategy for the other players' strategies $(a_1^*, \ldots, a_{i-1}^*, a_{i+1}^*, \ldots, a_I^*)$. In other words, no MD could further change its strategy to obtain smaller overhead, i.e., $Z_i(a_1^*, \ldots, a_{i-1}^*, a_i^*, a_{i+1}^*, \ldots, a_I^*) \leq Z_i(a_1^*, \ldots, a_{i-1}^*, a_i, a_{i+1}^*, \ldots, a_I^*)$, which is corresponding for $\forall a_i \in A_i$. We call the strategy $a^* = (a_1^*, \ldots, a_i^*, \ldots, a_I^*)$ is an NE of the game Γ.

After introducing the concept of NE, we will give the following NE property of potential game within finite improvements. The appealing property of potential game has been proved by *Theorem 1*.

Theorem 1: if the overhead minimization problem is formulated as a potential game by constructing a potential function which owns at least one NE within finite improvements.

Since potential game owns an NE within improvements, a sub-optimal computation offloading decision of overhead minimization problem can be achieved by designing a computing offloading algorithm based on the property of potential game. We will elaborate the PGOA in the following section.

C. Potential Game Based Offloading Algorithm

With the intention to get a satisfactory computation offloading decision in the system, PGOA is designed to achieve an NE (i.e.,sub-optimal solution). Specially, when MDs want to update their decisions, they will broadcast their parameters to all the MEC-MBSs. Then, PGOA allows one offloading MD which consumes minimal overhead to update its decision for each iteration. After many iterations, the offloading decisions of all the MDs will enter a stable status called as an NE of the overhead minimization problem. In this paper, the MD will make a computation offloading decision at each decision slot of a quasi-static scenario, denoted τ as one decision slot. When MDs update their decisions by applying PGOA, the process of making decisions mainly depends on the best response and better response rules. Accordingly, we introduce the concepts of better response and best response respectively.

Definition 4: The event where a player i changes a strategy decision from a_i to a_i', is a better response update if the following condition is satisfied, i.e.,

$$Z_i(a_i', a_{-i}) < Z_i(a_i, a_{-i}) \tag{16}$$

Definition 5: Given the strategies a_{-i} of the other players, when strategy set of one MD i changes until not have better response update any more, player i's strategy set $a_i^* \in A_i$ is called a best response if

$$Z_i(a_i^*, a_{-i}) < Z_i(a_i, a_{-i}) \tag{17}$$

The details of PGOA are briefly described in Algorithm 1. For PGOA, according to formula (16), during each given decision slot, all the MDs first send request-to-update (RTU) message. RTU represents each player wants to contend for the update decision (UD) opportunity to improve the current decision. RTU message which is judged by better response condition. If so, those MDs which send RTU message to servers will have opportunity to compete UD. Assume MD i wins the update decision, the MD would choose one best response UD with minimizing its overhead from the better response set. Then after a finite number of iterations, all MDs come to a mutual equilibrium state, which is the NE. Specially, how MDs update their decisions? We will list the operating process of PGOA from the following three steps.

First, MDs randomly initialize offloading decisions at a decision slot.

Then, at a given decision slot τ, compute the current optimal offloading decision a according to the interference and uplink data rate, deriving the better response set $\Theta(\tau)$. From set $\Theta(\tau)$, if the decision set of MD i satisfies $\Theta_i(\tau) \neq \varnothing$, the MD i will send RTU message to servers and contend for UD opportunity.

$$\Theta_i(\tau) = \{a_i' : Z_i(a_i', a_{-i}) < Z_i(a_i, a_{-i})\} \tag{18}$$

Finally, for the UD contention process, we will adopt the random back off-based mechanism to set the time τ between two broadcastings as a decision slot. Suppose MD i receives the information to update decision at a give time slot τ, to grantee the overhead minimization when offloaded, the MD will update its own decision from best response set $\Theta_i^*(\tau)$. Then, the MD will broadcast the message to all other MDs to indicate that it wins the UD opportunity. For MDs don't get the opportunity to update their decisions, they will hold on their original offloading decisions $a_i(\tau+1) = a_i(\tau)$. After finite number of iterations, all MDs come to a mutual Nash Equilibrium state.

$$\begin{aligned} \Theta_i^*(\tau) = \{a_i^* : Z_i(a_i^*, a_{-i}) < Z_i(a_i', a_{-i}), \\ a_i^* = arg \min_{a_i^*} Z_i(a_i', a_{-i}), a_i^* \in \Theta_i(\tau)\} \end{aligned} \tag{19}$$

Since the potential game always admits an NE within finite improvements, after multiple iteration times, for every MD,

Algorithm 1 Potential Game Based Offloading Algorithm

Initialization:
The strategy set of MD i: $a_i \in \{0, 1\}$;
Set initial decision slot: $\tau = 0$;
The number of decision slot: Ξ;
Each MD i's initial computation offloading decision $a_i(\tau) = 0$;
for each decisions slot $\tau \leq \Xi$ **do**
  compute the current wireless channel interference and the corresponding uplink data rate;
  if MD i is in decision slot τ **then**
    compute the threshold $C_{i,n}^m$ of each MD i in next slot τ and make a preliminary judgment;
    compute the better response $\Theta_i(\tau)$ of each MD i;
  end if
  while $\Theta_i(\tau) \neq \varnothing$ **do**
    each MD in $\Theta_i(\tau)$ send RTU message to servers and contend for UD opportunity;
    if MD i wins the UD opportunity **then**
      compute the best response of MEC i: $\Theta_i^*(\tau) \in \Theta_i(\tau)$;
      update a_i in $\Theta_i(\tau)$, i.e., $a_i(\tau + 1) = \Theta_i^*(\tau)$;
    else
      keep the offloading decision a_i unchanged: $a_i(\tau + 1) = a_i(\tau)$;
    end if
    for MDs j not wins the UD opportunities, they will keep decisions unchanged as well;
  end while
end for
until no RTU message in the MEC system.

there is not RTU message, i.e., $\Theta_i(\tau) = \emptyset$. It shows that the potential game reaches an NE, and the algorithm declares the end.

VI. Performance Analysis

A. Convergence Analysis of PGOA

The PGOA is convergent to an NE within finite improvement slots. During each decision slot, the iteration times of this algorithm is $O(I\Xi N)$. Suppose that the algorithm will be terminated after G iteration times, the total iteration times of PGOA are $O(GI\Xi N)$. However, the least iteration times need to be studied as well, since it is a measurement criteria of the worst-case performance guarantee of PGOA, we will do a simply analysis in the followings.

Assume that $C_{max} = \max_{n\in\mathbb{N}}\{C_{i,n}^m\}, C_{min} = \min_{n\in\mathbb{N}}\{C_{i,n}^m\}$. For PGOA, where $C_{i,n}^m$ is a non-negative non-integer, we obtain that the iteration times of PGOA are at least $\Xi * (\frac{1}{2}(INC_{\max})^2 + INC_{\max}C)/\Delta P_{max}$.

Proof: First, we can get that in terms of potential function

$$\begin{aligned} P(a) &\leq \frac{1}{2}\sum_{n=1}^{N}\sum_{i=1}^{I}\sum_{l=1}^{N}\sum_{j=1}^{I} C_{\max}^2 + \sum_{n=1}^{N}\sum_{i=1}^{I} C_{\max}C \\ &= \frac{1}{2}(INC_{\max})^2 + INC_{\max}C \end{aligned} \tag{20}$$

Lemma 3: When $0 \leq \Delta\kappa_1, \Delta\kappa_2, \Delta\kappa_3 \leq 1$, and $\sum_{j=1}^{3}\beta_j = 1$, the maximum difference ΔC of between two thresholds satisfy: $0 < \beta_1\Delta\kappa_1 + \beta_2\Delta\kappa_2 + \beta_3\Delta\kappa_3 \leq 2$.

Proof: See Appendix.

During a decision slot, suppose that one MD i updates its current decision a_i to the decision a_i', which will lead to a reduction of its objective function, $Z_i(a_i', a_{-i}) < Z_i(a_i, a_{-i})$. According to the definition of the potential game, we know that it also leads a reduction of the potential function by at most ΔP_{max}.

In order to prove this conclusion, the following three cases are considered.

For case 1): $a_i = 0, a_i' = 1$, according to formula (13), we know that

$$\begin{aligned} &P_i(a_i', a_{-i}) - P_i(a_i, a_{-i}) \\ &\quad = C_{i,n}^m \sum_{l\neq n}\sum_{j\neq i} C_{j,l}^m I_{\{a_j = a_i' : m\}} - C_{i,n}^m C \\ &\quad = C_{i,n}^m (\sum_{l\neq n}\sum_{j\neq i} C_{j,l}^m I_{\{a_j = a_i' : m\}} - C) \end{aligned} \tag{21}$$

According to *Lemma* 3, we know the $\Delta P_{\max}^1 = 2 \cdot C_{\max}$

For case 2): $a_i = 1, a_i' = 0$, the proof is similar to the proof of case 1, and $\Delta P_{\max}^2 = C_{\max} \cdot (|INC_{\max} - C|)$.

For case 3): $a_i = 1, a_i' = 1$, according to formula (15), we know that

$$\begin{aligned} &P_i(a_i', a_{-i}) - P_i(a_i, a_{-i}) \\ &= C_{i,n}^m \sum_{l\neq n}\sum_{j\neq i} C_{j,l}^m I_{\{a_j = a_i' : m\}} - C_{i,n}^m \sum_{l\neq n}\sum_{j\neq i} C_{j,l}^m I_{\{a_j = a_i : m\}} \\ &= C_{i,n}^m (\sum_{l\neq n}\sum_{j\neq i} C_{j,l}^m I_{\{a_j = a_i' : m\}} - \sum_{l\neq n}\sum_{j\neq i} C_{j,l}^m I_{\{a_j = a_i : m\}}) \end{aligned} \tag{22}$$

According to *Lemma* 3, we can also derive the $\Delta P_{\max}^3 = 2 \cdot IN \cdot C_{\max}$.

As a result, to grantee the worst-case performance, we value the maximization of ΔP, so we get $\Delta P_{\max} = \{\Delta P_{\max}^1, \Delta P_{\max}^2, \Delta P_{\max}^3\}$. With above, to get the NE of the PGOA, the iteration times of it will be not less than $\Xi * (\frac{1}{2}(INC_{max})^2 + INC_{max}C)/\Delta P_{max}$. Thus the worst-case performance guarantee is derived.

B. Efficiency Ratio of PGOA

With the analysis obtained above, we know that an NE is the minimum point of the potential function. If we formulate the objective function as one completely equivalent potential function in the potential game, the property of NE of the potential game is corresponding to the optimal offloading decision of the distributed computation offloading problem. In practice, the completely equivalent conditions are difficult to realize during the modeling process. So we decide to establish an approximate equivalent relation between potential function and objective function of the distributed computation offloading problem. The NE of potential game is regarded as one sub-optimal offloading decision of the distributed computation offloading problem. However, the efficiency of the sub-optimal offloading decision is still unknown, thus one index is

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

proposed to measure the approximate efficiency between the NE and the sub-optimal offloading decision [29], we introduce the definition of efficiency ratio (ER) in the game theory as follows.

We will measure the efficiency ratio where the overhead value of the overhead minimization problem is compared with the potential function value of the PGOA. Suppose all the MDs' strategies set at the equilibrium state is a^*. Then the ER is described as:

$$ER = \frac{\min \sum_{i=1}^{I} Z_i(a_i^*)}{\min \sum_{i=1}^{I} P_i(a_i^*)} \tag{23}$$

The closer to 1 the ER is, the better NE is. It shows that a^* is the sub-optimal computation offloading decision which realizes the objective of minimizing the overhead of the proposed problem.

VII. Simulations Results

In this section, we do simulations to evaluate the performance of the proposed distributed computation offloading algorithm. We consider the scenario with 50 MDs and 5 MEC-MBSs which cover $100m \times 100m$ area. The subchannel bandwidth is $W = 150kHz$. The transmission power is ranged from $p = 50mW$ to $150mW$ randomly, and the environment Gaussian noise is $N = -100dBm$. According to the wireless channel model for cellular radio environment, we set channel gain $g = l^{-\alpha}$, where α is the pass loss factor and we set $\alpha = 4$. The computation size b_i is randomly distributed between $0.5MB$ and $5MB$. We set $\eta_i \in [2000, 5000] cycles/MB$, which is random during the offloading process. For the purpose of simplifying the simulation, we assume that each MEC-MBS has the same computing ability $c_n = 5GHz$ and MDs' computing ability f_i^l is randomly distributed between $0.5GHz$ and $1GHz$. The energy consumption of one MEC-MBS for implementing a CPU cycle is $\varepsilon_0 = 1mJ/GHz$. For the decision weight of each MD i for the energy consumption and latency, λ_i^1, λ_i^2 is randomly assigned from the set $\{0, 0.5, 1\}$ and satisfies $\lambda_i^1 + \lambda_i^2 = 1$. For the parameters of threshold function C, based on our simulation results (see Fig.4), we set $\beta_1 = 0.4, \beta_2 = 0.3, \beta_3 = 0.3$.

We first present the convergence of the proposed computation offloading algorithm with the iteration times. As shown in Fig.2, the overhead of the PGOA decreases dramatically in the first 20 iteration times and then enters a stable status, where Fig.2 demonstrates that the proposed PGOA can lead potential game to one fix point after more than 20 iteration times. Because the potential game always admits an NE within the finite improvements, it is clear that the fix point is an NE point. With above, know that the the proposed computation algorithm is convergent.

Next, the overhead of the system with the different λ parameters is shown in Fig.3. By comparison, we can see as the total number of the MDs increases, the overhead of the system is growing. This is because when the total number of the MD rises, the interference between them becomes large, and the overhead of each MD becomes larger than that of before. Furthermore, we can see the overhead of the system

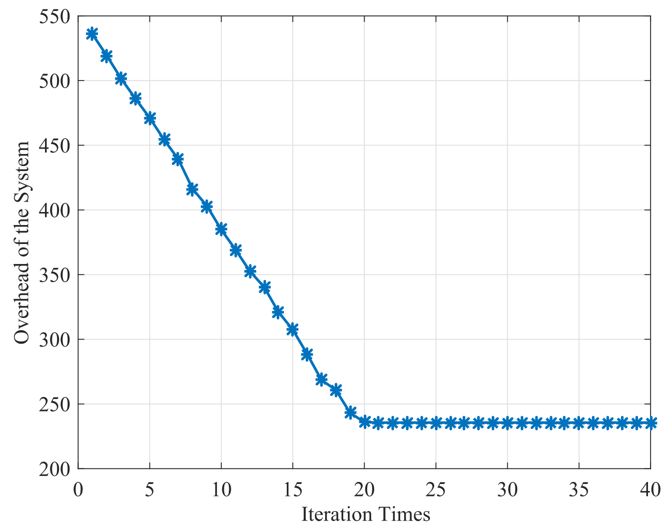

Fig. 2. The convergence of the potential game based offloading algorithm.

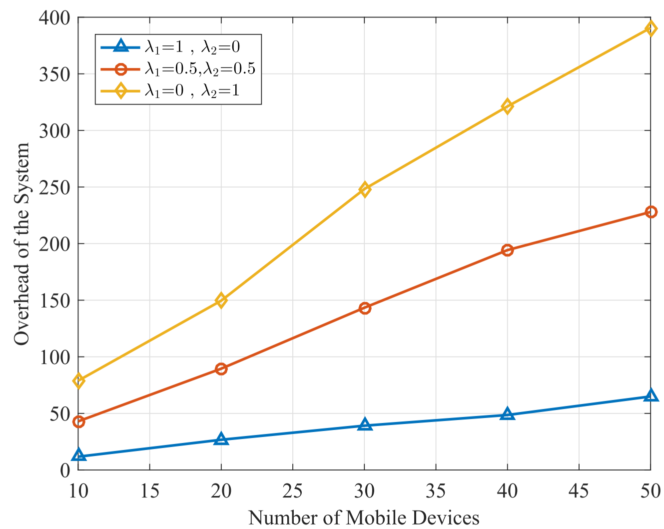

Fig. 3. Overhead of the system versus the different λ parameters.

$\lambda_1 = 0$ is significantly smaller than that of the system $\lambda_1 = 1$, which shows that the impact of the energy consumption is far higher than that of delay for total overhead of the system.

Fig.4 shows the overhead of the system with the different parameters of β with the threshold function C. Note that the larger the value of parameter β_1, the smaller the overhead of the system. Especially, when the value of parameter β_1 reaches 0.4, the change of the value of parameter β_2 has not significantly influence on the overhead of the system. As can be observed from the red curve and green curve, the gap between the cyan curve and the blue curve is not wide. Besides, we can also see that the larger the value of parameter β_3, the larger the overhead of the system, which shows energy consumption occupies the main effect on the overhead of the system. Accordingly, to guarantee the saving overhead efficiency of our designed algorithm, we set $\beta_1 = 0.4, \beta_2 = 0.3, \beta_3 = 0.3$ as the weighting parameters of β with the threshold function C combined with our research scenario.

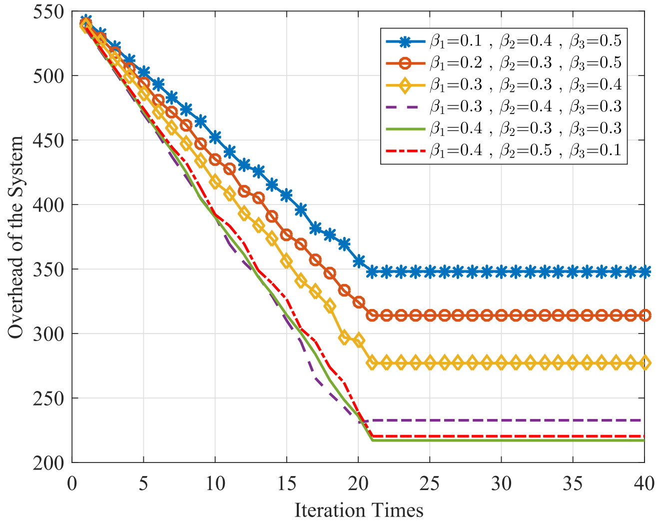

Fig. 4. Overhead of the system versus the different parameters of threshold C.

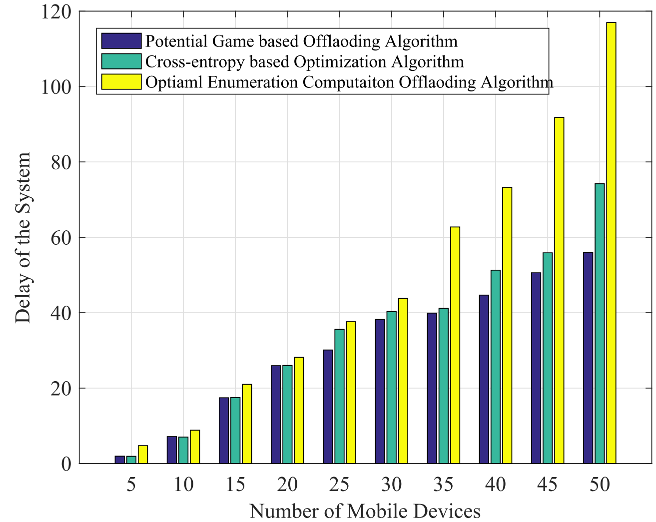

Fig. 6. Delay of the system versus the different number of mobile devices.

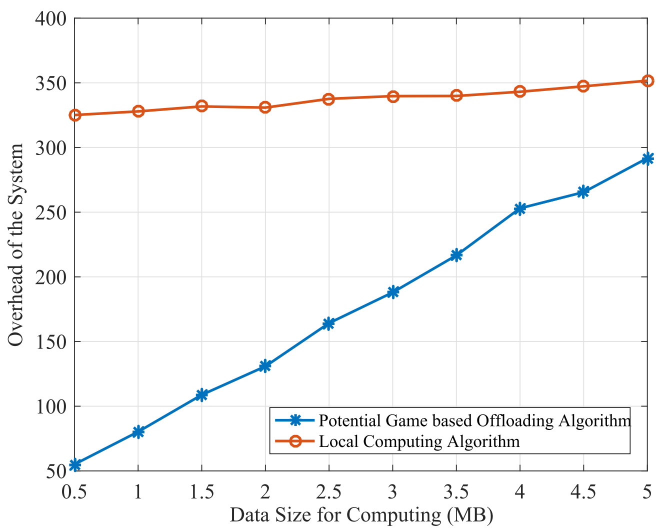

Fig. 5. Overhead of the system versus different data sizes for computing.

In Fig.5, we investigate the overhead of the system with different data sizes for computing. We implement the simulations with the data size $0.5 \sim 5$MB, the overhead of PGOA is compared with that of the local computing algorithm. As we can see, the date size is proportional to the computation resource by factor η_i for each MD i, so the change of the data size will affect the computation resource consumption, energy consumption, communication and computation delay. Fig.5 shows that with data size for computing rising, the overhead of the local computing algorithm increases slowly. That is because there is not transmission energy consumption and delay increase for the local computing algorithm, the overhead of the system mainly is affected by the computation delay when computing locally. However, we observe that the overhead of PGOA increases dramatically with the rising of the data size for computing. It is due to that the increased overhead of the system, not only contains the computation delay, but also contains the communication energy consumption and delay, which is a mainly influence factor for overhead of the system compared with the local computing algorithm. As a whole, although the overhead of PGOA is fast-growing, the total overhead of PGOA is lower than that of local computing algorithm within $5MB$. With above, know that our designed algorithm is efficient and practical for saving overhead of the system.

In Fig.6, the delay of PGOA is compared with the other two algorithms, cross-entropy based optimization algorithm [30] and optimal enumeration computation offloading algorithm [31] respectively. Simulation results show that with the number of the MD growing, the delay of all the algorithms increase dramatically. One reason is that with more and more MDs joining into the system, the interference between offloading MDs becomes serious, so the transmission delay of the system is increasing, simultaneously, the other reason is that since more and more MDs need to share the computation resource of one MEC-MBS, the transmission delay is also growing. Furthermore, as depicted in Fig.6, PGOA is confirmed to have much higher computational efficiency than other two algorithms. It is because the cross-entropy optimization algorithm is centralized, which has the whole information collection process, the delay of cross-entry optimization algorithm is higher than that of PGOA. Fig.6 shows the delay of the system of PGOA can save delay at most 11% compared with the cross-entropy based optimization algorithm. Finally, for the optimal enumeration computation offloading algorithm, which is also a centralized optimization algorithm, so it is no wonder that the computational delay performance of PGOA is highest. With above, our designed algorithm is efficient for saving delay of the system.

We also do the simulation for the overhead saving efficiency of PGOA, cross-entropy based optimization algorithm [30] and optimal enumeration computation offloading algorithm [31] in Fig.7. To show the overhead saving performance, we implement the simulations with the overhead of the three algorithms varying with the number of the MD. In this case, firstly, we compare the overhead of PGOA with that of the

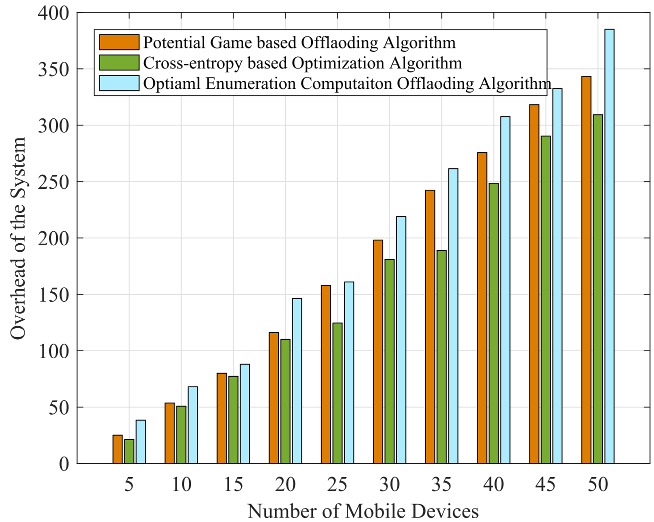

Fig. 7. Overhead of the system versus the different number of mobile devices.

cross-entropy based optimization algorithm. One can observe that the gap between them is not wide, and the cross-entropy based optimization algorithm can save overhead 18% compared with PGOA. Notice that the reason of this phenomenon is mainly that the difference of processing ways between distributed computation offloading method and centralized computation offloading method. Specially, the centralized computation offloading methods increase the delay of massive information collection, but the whole energy consumption of computational process is reduced. However, the PGOA which is a distributed computation offloading method owns those advantages: First, compared with other centralized computation offloading algorithm, the design philosophy of PGOA is to satisfy different individuals and interests. Simultaneously, in terms of NE characteristic of potential game, PGOA can obtain the self-stability to reach an NE within finite improvements. Accordingly, from the aspects of paying more attention to the individual needs, sacrificing some network performance to satisfy more MDs' individual interests is worthy.

VIII. Conclusion

In this paper, we consider the multi-device and multi-MEC scenario of a SCN integrated with MEC. To pursue individual computation demands and guarantee network performance, we formulate a distributed overhead minimization problem, which minimizes the overhead of each MD, so a potential game theory is adopted to prove our proposed strategy game is a potential game problem. Then by feat of the excellent convergence of potential game to achieve an NE within finite improvements, we develop a PGOA. In addition, to measure the computational performance of our proposed algorithm, we analyze the least iteration times of the PGOA to obtain the worst-case performance guarantee and the efficiency ratio metric. Finally, compared with other existing computation algorithms, the simulation results show that the PGOA can not only save offloading delay, but also guarantee the saving overhead efficiency with the maximum extent when permitting each MD to pursue individual interests.

Appendix
Proof of Lemma 3

Lemma 3: When there is $0 \leq \Delta\kappa_1, \Delta\kappa_2, \Delta\kappa_3 \leq 1$*, and* $\sum_{j=1}^{3} \beta_j = 1$*, the maximum difference* ΔC *of between two thresholds satisfies:* $0 < \beta_1\Delta\kappa_1 + \beta_2\Delta\kappa_2 + \beta_3\Delta\kappa_3 \leq 2$.

Proof: Suppose that arbitrary MD i's current decision is a_i, we can get $C_{i,n}^m = \beta_1\kappa_1^i + \beta_2\kappa_2^i + \beta_3\kappa_3^i$ when the mobile device changes the current decision a_i into a_i', we can obtain $C_{i,j}^m = \beta_1\kappa_1^{i\prime} + \beta_2\kappa_2^{i\prime} + \beta_3\kappa_3^{i\prime}$, where $\beta_1 + \beta_2 + \beta_3 = 1$, and $0 < \kappa_1^i, \kappa_2^i, \kappa_3^i, \kappa_1^{i\prime}, \kappa_2^{i\prime}, \kappa_3^{i\prime} \leq 1$. According to the analysis above, we can get $0 \leq \Delta\kappa_1 = \kappa_1^i - \kappa_1^{i\prime} \leq 1$, the same $0 \leq \Delta\kappa_2, \Delta\kappa_3 \leq 1$. As a result, we can also obtain $0 < \beta_1\Delta\kappa_1 + \beta_2\Delta\kappa_2 + \beta_3\Delta\kappa_3 \leq 2$.
According to difference of two squares, we know that:

$$\begin{aligned}\beta_1\Delta\kappa_1 &+ \beta_2\Delta\kappa_2 + \beta_3\Delta\kappa_3 \\ &\leq \frac{{\beta_1}^2 + {\Delta\kappa_1}^2}{2} + \frac{{\beta_2}^2 + {\Delta\kappa_2}^2}{2} + \frac{{\beta_3}^2 + {\Delta\kappa_3}^2}{2} \\ &= \frac{{\Delta\kappa_1}^2 + {\Delta\kappa_2}^2 + {\Delta\kappa_3}^2 + {\beta_1}^2 + {\beta_2}^2 + {\beta_3}^2}{2} \end{aligned} \tag{24}$$

According to square formula, we know that:

$$\begin{aligned}{\beta_1}^2 &+ {\beta_2}^2 + {\beta_3}^2 \\ &= (\beta_1 + \beta_2 + \beta_3)^2 - 2\beta_1\beta_2 - 2\beta_2\beta_3 - 2\beta_1\beta_3 \end{aligned} \tag{25}$$

The known condition is as follows:$\beta_1 + \beta_2 + \beta_3 = 1$. Thus

$${\beta_1}^2 + {\beta_2}^2 + {\beta_3}^2 = 1 - 2\beta_1\beta_2 - 2\beta_2\beta_3 - 2\beta_1\beta_3 \tag{26}$$

Since $\beta_1, \beta_2, \beta_3$ satisfies $0 \leq \beta_1, \beta_2, \beta_3 \leq 1$, and $\Delta\kappa_1, \Delta\kappa_2, \Delta\kappa_3$ satisfies $0 \leq \Delta\kappa_1, \Delta\kappa_2, \Delta\kappa_3 \leq 1$,
We can conclude that:

$$\begin{aligned} &0 \leq \beta_1\beta_2 \leq 1, \quad 0 \leq \beta_2\beta_3 \leq 1, \quad 0 \leq \beta_1\beta_3 \leq 1, \\ &\qquad {\beta_1}^2 + {\beta_2}^2 + {\beta_3}^2 \leq 1, \quad 0 \leq \Delta\kappa_1^2, \Delta\kappa_2^2\Delta\kappa_3^2 \leq 1. \end{aligned} \tag{27}$$

Based on the analysis and results, the maximum difference ΔC of between two thresholds indeed satisfies:

$$0 < \beta_1\Delta\kappa_1 + \beta_2\Delta\kappa_2 + \beta_3\Delta\kappa_3 \leq 2 \tag{28}$$

This completes the proof.

References

[1] S. Wang, R. Urgaonkar, M. Zafer, T. He, K. Chan, and K. K. Leung, "Dynamic service migration in mobile edge-clouds," in *Proc. IFIP Netw. Conf.*, May 2015, pp. 1–9.

[2] Z. Fu, K. Ren, J. Shu, X. Sun, and F. Huang, "Enabling personalized search over encrypted outsourced data with efficiency improvement," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 9, pp. 2546–2559, Sep. 2016.

[3] 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.

[4] J. Wu, C. Yuen, N.-M. Cheung, J. Chen, and C. W. Chen, "Enabling adaptive high-frame-rate video streaming in mobile cloud gaming applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 12, pp. 1988–2001, Dec. 2015.

[5] A. U. R. Khan, M. Othman, S. A. Madani, and S. U. Khan, "A survey of mobile cloud computing application models," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 393–413, 1st Quart., 2013.

[6] M. Kamel, W. Hamouda, and A. Youssef, "Ultra-dense networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2522–2545, 4th Quart., 2016.

[7] S. Chen, R. Ma, H.-H. Chen, H. Zhang, W. Meng, and J. Liu, "Machine-to-machine communications in ultra-dense networks—A survey," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1478–1503, 3rd Quart., 2017.

[8] S. Samarakoon, M. Bennis, W. Saad, M. Debbah, and M. Latva-Aho, "Ultra dense small cell networks: Turning density into energy efficiency," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1267–1280, May 2016.

[9] T. G. Rodrigues, K. Suto, H. Nishiyama, and N. Kato, "Hybrid method for minimizing service delay in edge cloud computing through VM migration and transmission power control," *IEEE Trans. Comput.*, vol. 66, no. 5, pp. 810–819, May 2017.

[10] Y. Cao, C. Long, T. Jiang, and S. Mao, "Share communication and computation resources on mobile devices: A social awareness perspective," *IEEE Wireless Commun.*, vol. 23, no. 4, pp. 52–59, Aug. 2016.

[11] 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.

[12] X. Sun and N. Ansari, "EdgeIoT: Mobile edge computing for the Internet of Things," *IEEE Commun. Mag.*, vol. 54, no. 12, pp. 22–29, Dec. 2016.

[13] M. Randles, D. Lamb, and A. Taleb-Bendiab, "A comparative study into distributed load balancing algorithms for cloud computing," in *Proc. IEEE Int. Conf. Adv. Inf. Netw. Appl. Workshops*, Apr. 2010, pp. 551–556.

[14] A. M. Khan, X. Vilaca, L. Rodrigues, and F. Freitag, "A distributed auctioneer for resource allocation in decentralized systems," in *Proc. IEEE 36th Int. Conf. Distrib. Comput. Syst.*, Aug. 2016, pp. 201–210.

[15] D. Wu, Q. Wu, Y. Xu, and Y.-C. Liang, "QoE and energy aware resource allocation in small cell networks with power selection, load management, and channel allocation," *IEEE Trans. Veh. Technol.*, vol. 66, no. 8, pp. 7461–7473, Aug. 2017.

[16] 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.

[17] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE Trans. Signal Inf. Process. Over Netw.*, vol. 1, no. 2, pp. 89–103, Jun. 2015.

[18] K. Zhang, S. Leng, Y. He, S. Maharjan, and Y. Zhang, "Mobile edge computing and networking for green and low-latency Internet of Things," *IEEE Commun. Mag.*, vol. 56, no. 5, pp. 39–45, May 2018.

[19] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2795–2808, Oct. 2016.

[20] T. Zhao, S. Zhou, X. Guo, Y. Zhao, and Z. Niu, "A cooperative scheduling scheme of local cloud and Internet cloud for delay-aware mobile cloud computing," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2015, pp. 1–6.

[21] L. Yang, H. Zhang, M. Li, J. Guo, and H. Ji, "Mobile edge computing empowered energy efficient task offloading in 5G," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6398–6409, Jul. 2018.

[22] L. Tianze, W. Muqing, Z. Min, and L. Wenxing, "An overhead-optimizing task scheduling strategy for ad-hoc based mobile edge computing," *IEEE Access*, vol. 5, pp. 5609–5622, 2017.

[23] W. Zhang, Y. Wen, and H. H. Chen, "Toward transcoding as a service: Energy-efficient offloading policy for green mobile cloud," *IEEE Netw.*, vol. 28, no. 6, pp. 67–73, Nov. 2014.

[24] K. Zhang *et al.*, "Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks," *IEEE Access*, vol. 4, pp. 5896–5907, 2016.

[25] N. Sun, Y. Zhao, L. Sun, and Q. Wu, "Distributed and dynamic resource management for wireless service delivery to high-speed trains," *IEEE Access*, vol. 5, pp. 620–632, Jan. 2017.

[26] K. Xie *et al.*, "Distributed multi-dimensional pricing for efficient application offloading in mobile cloud computing," *IEEE Trans. Services Comput.*, to be published.

[27] S. Bi and Y. J. Zhang, "Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading," *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 4177–4190, Jun. 2018.

[28] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 974–983, Apr. 2015.

[29] M. Voorneveld, "Equilibria and approximate equilibria in infinite potential games," *Econ. Lett.*, vol. 56, no. 2, pp. 163–169, Oct. 1997.

[30] C. Izack, G. Boaz, and S. Avraham, "Managing stochastic, finite capacity, multi-project systems through the cross-entropy methodology," *Ann. Oper. Res.*, vol. 134, no. 1, pp. 183–199, Feb. 2005.

[31] H. Guo, J. Liu, and H. Qin, "Collaborative mobile edge computation offloading for IoT over fiber-wireless networks," *IEEE Netw.*, vol. 32, no. 1, pp. 66–71, Jan. 2018.

Lichao Yang received the B.S. degree in mathematics and statistics from Zhengzhou University, China, in 2015. She is currently pursuing the Ph.D. degree with the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing, China. She is currently a Visiting Ph.D. Student with The University of Texas at Arlington, USA. Her current research interests include 5G small-cell networks, fog and edge computing, and online multiple-dimension resource management.

Heli Zhang received the B.S. degree in communication engineering from Central South University in 2009 and the Ph.D. degree in communication and information system from the Beijing University of Posts and Telecommunications (BUPT) in 2014. Since 2014, she has been a Lecturer with the School of Information and Communication Engineering, BUPT.

Her research interests include heterogeneous networks, long-term evolution/fifth generation, and Internet of Things. She is a Reviewer for the journals of the IEEE WIRELESS COMMUNICATIONS, the *IEEE Communication Magazine*, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the IEEE COMMUNICATION LETTERS, and the IEEE TRANSACTIONS ON NETWORKING. She participated in many national projects funded by the National Science and Technology Major Project, the National 863 High-Tech, and the National Natural Science Foundation of China, and cooperated with many corporations in research.

Xi Li received the B.E. and Ph.D. degrees in the major of communication and information system from the Beijing University of Posts and Telecommunications (BUPT) in 2005 and 2010, respectively.

In 2017, she was a Visiting Scholar with The University of British Columbia, Vancouver, BC, Canada. She is currently an Associate Professor with the School of Information and Communication Engineering, BUPT. She has authored over 100 papers in international journals and conferences. Her current research interests include resource management and networking in the next-generation networks, cloud computing, edge caching, and mobile Internet. She is serving on the editorial boards for the *Wiley International Journal of Communication Systems*. She also served as the Chair for the special track on cognitive testbed in the Chinacom 2011, as a TPC Member for the IEEE WCNC 2012/2014/2015/2016/2019, the PIMRC 2012/2017/2018, the Globecom 2015/2017/2018, the ICC 2015/2016/2017/2018/2019, the Infocom 2018, and the Cloud-Com 2013/2014/2015, and a peer reviewer for many academic journals.

Hong Ji (SM'09) received the B.S. degree in communications engineering and the M.S. and Ph.D. degrees in information and communications engineering from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 1989, 1992, and 2002, respectively. In 2006, she was a Visiting Scholar with The University of British Columbia, Vancouver, BC, Canada. She is currently a Professor with BUPT. She has authored over 200 journal/conference papers. Several of her papers had been selected for best paper. Her research interests are in the areas of wireless networks and mobile systems, including green communications, radio access, ICT applications, system architectures, cloud computing, software-defined networks, management algorithms, and performance evaluations. She is serving on the editorial boards for the IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING and the *Wiley International Journal of Communication Systems*. She guest edited the *Wiley International Journal of Communication Systems*, Special Issue on Mobile Internet: Content, Security and Terminal. She served as the Co-Chair for Chinacom11, and as a member for the Technical Program Committee of INFOCOM18 Workshops, ICC18 Workshops, PIMRC18/17, ISCIT17, GC17 Workshops, Globecom16/15/14/13/12/11, ICC13/12/11, IEEE VTC12S, and WCNC15/12.

Victor C. M. Leung (S'75–M'89–SM'97–F'03) received the B.A.Sc. degree (Hons.) in electrical engineering from The University of British Columbia (UBC) in 1977 and the Ph.D. degree in electrical engineering from UBC in 1982. He was a recipient of the APEBC Gold Medal as the Head of the 1977 graduating class with the Faculty of Applied Science, UBC, and the Canadian Natural Sciences and Engineering Research Council Postgraduate Scholarship from 1977 to 1981.

From 1981 to 1987, he was a Senior Member of Technical Staff and a Satellite System Specialist with MPR Teltech Ltd., Canada. In 1988, he was a Lecturer with the Department of Electronics, The Chinese University of Hong Kong. In 1989, he returned to UBC as a Faculty Member, where he is currently a Professor and the TELUS Mobility Research Chair in Advanced Telecommunications Engineering with the Department of Electrical and Computer Engineering. He has co-authored over 1200 journal articles, conference papers, and book chapters, and co-edited 14 book titles. Several of his papers had been selected for best paper awards. His research interests are in the broad areas of wireless networks and mobile systems.

Dr. Leung has co-authored papers that won the 2017 IEEE ComSoc Fred W. Ellersick Prize, the 2017 IEEE Systems Journal Best Paper Award, and the 2018 IEEE CSIM Best Journal Paper Award. He is a registered Professional Engineer in BC, Canada. He is a fellow of the Royal Society of Canada, The Engineering Institute of Canada, and The Canadian Academy of Engineering. He was a Distinguished Lecturer of the IEEE Communications Society. He was a recipient of the IEEE Vancouver Section Centennial Award, the 2011 UBC Killam Research Prize, the 2017 Canadian Award for Telecommunications Research, and the 2018 IEEE TGCC Distinguished Technical Achievement Recognition Award. He is serving on the editorial boards for the IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING, the IEEE TRANSACTIONS ON CLOUD COMPUTING, the IEEE ACCESS, the IEEE NETWORK, *Computer Communications*, and several other journals, and has previously served on the editorial boards for the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS–Wireless Communications Series and Series on Green Communications and Networking, the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the IEEE TRANSACTIONS ON COMPUTERS, the IEEE WIRELESS COMMUNICATIONS LETTERS, and the *Journal of Communications and Networks*. He guest edited many journal special issues and provided leadership to the organizing committees and technical program committees of numerous conferences and workshops.