Delay and Energy Aware Computation Task Offloading Strategy in Power Wireless Heterogeneous Networks

Wei Bai[1,2(✉)], Yang Lu[1,2], Donglei Zhang[1,2], Jawei Li[3], Ping Ma[4], Shuiyao Chen[5], and WeiPing Shao[5]

[1] Electric Power Intelligent Sensing Technology and Application State Grid Corporation Joint Laboratory, Beijing, China
baiwei@geiri.sgcc.com.cn
[2] Global Energy Interconnection Research Institute Co., Ltd., Beijing, China
[3] Beijing University of Posts and Telecommunications, Beijing, China
[4] State Grid Shaoxing Electric Power Supply Company Shaoxing, Shaoxing, China
[5] State Grid Zhejiang Electric Power Co., Ltd., Hangzhou, China

Abstract. Mobile Edge Computing (MEC) is regarded as a promising technology that migrates cloud computing platforms with computing and storage capabilities to the edge of the wireless access network, enabling rich applications and services in close proximity to the mobile users (MUs). There are a lot of literatures that have studied computation offloading. Different from them, this paper performs a novel research on multi-level computation offloading taking account into the heterogeneity of computation tasks and computation resource backup pool, and introducing opportunistic networks in multi-access networks simultaneously. Firstly, we describe the computation offloading model. Then, we formulate the multi-level computation offloading problem as a Stackelberg game and demonstrate the existence of the game Nash equilibrium. In order to solve above problem, we design a global optimal algorithm based on game theory. Finally, the performance of the proposed algorithm is verified by comparing with other algorithms. Simulation results corroborate that the algorithm can not only decrease the energy consumption, but also is stable.

Keywords: Mobile edge computing · Stackelberg game · Computation offloading · Multi-access network

1 Introduction

Nowadays, the development of latency sensitive and computation intensive applications such as interactive game and augmented reality is closely related to the development of mobile devices. These applications have attracted a lot of attention because of their ability to spice up and bring convenience to our lives. Nevertheless, mobile devices are often resource constrained [1]. As a result, resource-constrained mobile devices can become strained by latency sensitive and computation intensive applications, causing bottlenecks and making it difficult for

S. W. Loke et al. (Eds.): MONAMI 2020, LNICST 338, pp. 94–110, 2020.
https://doi.org/10.1007/978-3-030-64002-6_7

mobile devices to ensure the required quality of service (QoS) (including energy consumption and/or processing delay [2]. Mobile Edge Computing (MEC) proposed by ETSI (European Telecommunications Standards Institute) can mitigate the need for large amounts of computation from mobile devices, which enables such applications through providing computation resource in close to mobile users and at the edge of the radio access network [3].

Computation offloading is one of the critical issues in MEC system. Generally speaking, computation offloading is made up of three phases, the data uploading, the task execution and the result return. However, since the computation results are very small, the time for returning the computation results is ignored in most literatures [4]. With computation offloading technology, the resource-constrained MUs can save energy and enrich MUs experience by fully or partially offloading computation-intensive tasks to the nearby other MU or MEC server [5].

There are a lot of work that has investigated the computation offloading and resource allocation of MEC. In a multi-user scenario, paper [5] investigated the resource allocation for MEC offloading in cellular networks. Considering a multi-user and multi-server MEC scenario, the authors in [6] proposed a genetic algorithm based computation algorithm in order to solve offloading decision, computation resource allocation and channel allocation. As indicated in [7], the efficiency of the computation offloading and resource allocation depends largely on the decision to offloading strategy. Task offloading and resource allocation in an MEC enabled multi-cell wireless network are studied in [8] with the objective of maximizing the MUs' task offloading gains and it formulate considered problem as a mixed integer nonlinear program. However, the authors did not consider the heterogeneity of MUs computation tasks and the queue latency at the MEC servers. Various from above work, we do a novel research on the computation offloading decision-making and resource allocation in a multi-access network.

In this paper, we study a MEC system in which consider computation resource backup pool among the MEC servers and investigate computation offloading strategy with the objective of maximizing the utility of MUs and the revenue of MEC servers via optimizing offloading decision-making while taking into account heterogeneity of computation tasks, queue delay and opportunistic networks (ONs) in multi-access networks, where, opportunistic networks is a self-organizing network which introduces encountering opportunities brought by node movement to realize communication rather than relying on a complete link between the source node and the target node [9] and the ONs enable MUs to communicate not only with MUs connected by D2D link, but also with computing nodes via movement. First, we denote the system model and formulate the main problem as a Stackelberg game problem. Then, we prove the existence of the game Nash equilibrium. Furthermore, we propose a global optimal algorithm namely computation offloading algorithm to tackle this problem. Eventually, the convergence of the algorithm is studied by simulation. The simulation results show that the algorithm works well comparing with other algorithms, which verifies the effectiveness of the proposed algorithm in terms of maximizing the utility of MUs and the revenue of MEC servers.

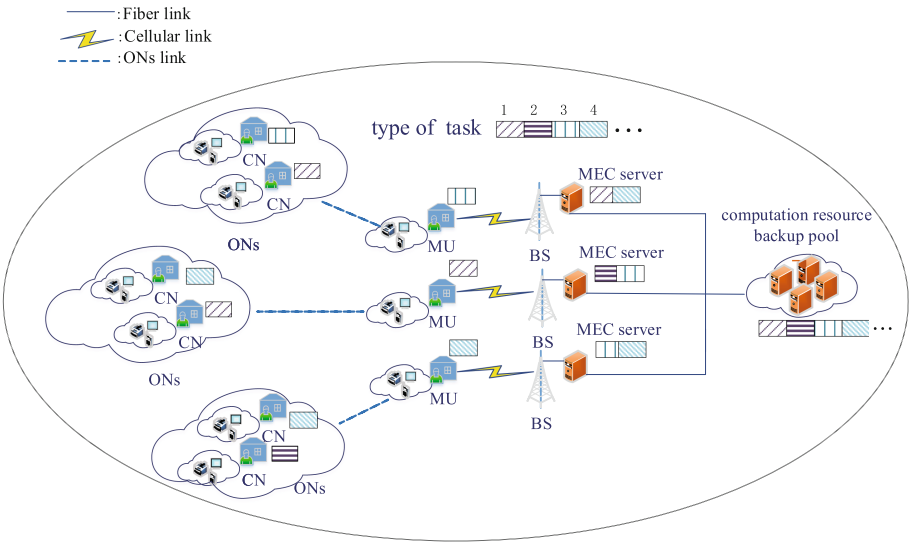

Fig. 1. System model

The remainder of our work is organized as follows. Section 2 gives the system model and problem formulation. In Sect. 3, the proposed joint optimization algorithm is developed. Simulation results and discussions are described in Sect. 4. Finally, we conclude this paper in Sect. 5.

2 System Model

2.1 Network Model

We consider a multi-access network with I mobile users (MUs) and M small base stations (SBSs) as shown in Fig. 1, where all of SBSs is equipped with MEC servers. The set of MUs and MEC servers are denoted by $I = \{1, 2, ..., i, ..., I\}$ and $N = \{1, 2, ..., k, ..., N\}$, in which i and k represent MU i and MEC server k, respectively. Moreover, we suppose that each MU has at most M types of computation tasks to be accomplished, and each task cann't be divided into. Also, the set of computation task types requested by MUs is denoted by $M = \{1, 2, ..., j, ..., M\}$, which include interactive games, face/fingerprint recognition, natural language processing, and so on. In addition, we introduce $p_{i,j}$ to indicate the request probability of the type-j task at MU i taking into account that different computation tasks are randomly requested by MUs at a time, where $0 < p_{i,j} < 1$. As for type-j computation task at MU i, it can be described as three items, i.e., $b_{i,j} = \{L_{i,j}, D_{i,j}, T_{i,j}^{max}\}$, in which $L_{i,j}$ denotes the data size of computation task, $D_{i,j}$ describes the total number of CPU cycles required to complete the computation task measured in Megacycle, $T_{i,j}^{max}$ represents the Quality of Service (QoS) requirement of computation task. Furthermore, the

maximum computation capacity of MU i and MEC server k is separately $f_i^{l,max}$ and $f_k^{e,max}$.

In the MEC system, between MU i and MEC server k are connected with cellular link, and each of MUs is capable of communicating with other MUs nearby via opportunistic networks (ONs), which communicates using short-range communication protocols, for example, wifi, bluetooth, and etc. For ease of description, the set of MUs in close to the MU i is described as U and called help users (HUs). Otherwise, let $\delta_{i,j}$ denote offloading decision for type-j computation task at MU i, $\delta_{i,j} = \{-I, -(I-1), ..., -m, -(i+1), -(i-1), ..., 0, 1, 2, ..., k, ...N\}$, $m \in I \backslash i$, $k \in N$. Specifically, $\delta_{i,j} = -m$ if MU i offloads type-j computation task to HUs m, $\delta_{i,j} = 0$ if MU i decides to execute type-j computation task locally by its own CPU, $\delta_{i,j} = k$ if MU i offloads type-j computation task to MEC server k. Next, we will describe the processing delay and energy consumption in the above mentioned cases.

2.2 Local Computing

For $\delta_{i,j} = 0$, MU i chooses to execute type-j computation task by its own CPU. Suppose that the computation capacity of each MU is fixed when computing, but may vary among the MUs. Denote the available computation capacity of MU i as $f_i^l = a_i^l f_i^{l,max}$, $a_i^l \in (0, 1]$, which is measured by CPU cycles per second. Therefore, the total delay in this case can be expressed by

$$T_{i,j}^l = \frac{D_{i,j}}{f_i^l} \tag{1}$$

Given e_i^l which denotes the consumed energy of MU i per CPU cycle for local computing, the energy consumption in such case can be given by

$$E_{i,j}^l = D_{i,j} e_i^l \tag{2}$$

2.3 Offload to MEC Servers

For $\delta_{i,j} = k$, MU i decides to offload type-j computation task to MEC sever k. Considering the case where multiple MUs are simultaneously offloaded to the same MEC server k, although the MEC server can purchase computation resources from the network operator, which places a computation resource backup pool (CRBP) in each small cell, there is a queue delay because the purchase of resources requires a certain amount of cost, and the MEC server is rational. In order to compute execution time and queue time of computation task $b_{i,j}$ at the MEC server k, we suppose that the probability of computation task requests at MU i follows a Poisson distribution, and then the arrival of various types of computation tasks at the MEC server k is a Poisson process. After that, we can model the execution and arrival of computation tasks at the MEC server as an $M/M/1$ queue. Therefore, the service time of computation tasks at the MEC server (including execution time and queuing time) follows an exponential

distribution with an average service time of $\frac{1}{\sigma^M}$, where $\sigma^M = f_k^{e,max} + f_k^{e,c}$, $f_k^{e,c}$ is the computation resources purchased from network operator. Hereafter,the average time to complete computation task $b_{i,j}$ in this case can be denoted by $t_{i,j} = \frac{D_{i,j}}{\sigma^M - \overline{\lambda_M}}$. Here, $\overline{\lambda_M}$ is the average arrival rate of the computation task $b_{i,j}$ at the MEC server, it can be described as:

$$\overline{\lambda_M} = \sum_{i=1}^{I} \sum_{j=1}^{M} p_{i,j} D_{i,j} X_{\{\delta_{i,j}=k\}} \tag{3}$$

where, $X_{\{\#\}} = 0$ if the condition # is false, and otherwise $X_{\{\#\}} = 1$. Furthermore, the total delay of computation task $b_{i,j}$ in the case can be calculated by

$$T_{i,j}^k = \frac{L_{i,j}}{R_{i,j}^k} + \frac{D_{i,j}}{\sigma^M - \overline{\lambda_M}} \tag{4}$$

Here, $R_{i,j}^k$ represents the data transfer rate from the MU i to the MEC server k. Moreover, let $P_{i,j}$, $h_{i,k}$ and B_k denote the transmission power of MU i, the channel gain and the channel bandwidth, respectively. $R_{i,j}^k$ can be computed by

$$R_{i,j}^k = B^k \log_2 \left(1 + \frac{P_{i,j} |h_{i,k}|^2}{N_0 + \sum\limits_{i' \in U \setminus i, j' \in M \setminus j, \delta_{i,j}=1} P_{i',j'} |h_{i',k}|^2} \right) \tag{5}$$

where N_0 is the additive white gaussian noise power. Thus, the total energy consumption for offloading to MEC server can be denoted by

$$E_{i,j}^k = P_{i,j} T_{i,j}^k \tag{6}$$

2.4 Offload to Help Users

For $\delta_{i,j} = -m$, MU i decides to offload type-j computation task to help user m. Since there exists a large queue delay for MU i when it decides to offload computation task $b_{i,j}$ to the MEC server during peak hours, the MUs can also choose to offload to a help user with a large amount of idle resources nearby. In this case, the MU i transmits the computation task $b_{i,j}$ to the help user m through the ONs. After that, we assume that each help user can only help one user at a time taking into account the computation resources limitation of MUs', and then the transmission rate can be expressed as

$$R_{i,j}^m = B^m \log_2 \left(1 + \frac{P_{i,j} |h_{i,m}|^2}{N_0} \right) \tag{7}$$

where B^m and $h_{i,m}$ denote the channel bandwidth and the channel gain between MUi and help user m. Let f_i^m represents available computation resource of help user m, the total execution delay can be given by

$$T_{i,j}^m = \frac{L_{i,j}}{R_{i,j}^m} + \frac{D_{i,j}}{f_i^m} \tag{8}$$

Therefore, the total energy consumption in such case can be expressed as

$$E_{i,j}^{m} = P_{i,j} T_{i,j}^{m} \tag{9}$$

3 Problem Formulation and Proposed Algorithm

3.1 Problem Formulation

Considering that the MUs and MEC servers are always maintained and operated by some operators, in order to gain revenues from the provision of computation services, the operator adopts a pricing scheme so that the MUs can be charged according to the computation resources they require. In addition, we consider the linear cost function MEC servers that provide resources. The cost of MEC server k that provides x unit resources for computation offload is expressed as

$$C_k(x) = \beta_k^e x + \gamma_k^e \tag{10}$$

where $\beta_k^e > 0$ and γ_k^e represent the coefficients of cost function for MEC server. For ease of analysis, we have $\gamma_k^e = 0$.

Due to the computation resource limitation of the MEC server, there may be competition among the MUs when making offloading decision. Since the MUs' offloading decisions are driven by the pricing scheme, these MUs are indirectly coupled by the computation resource price. Moreover, the computation resource providers, i.e., the MEC servers, are also indirectly coupled through the price during the offloading process.

The MUs can offload the computation task to the help users nearby or the MEC servers remotely, and the MUs' offloading decisions are response to the price advertised by the MEC servers. Therefore, the Stackelberg game is an attractive method to model the multi-level computation offloading [13]. In this game, the MEC server is the leader, and the MUs act as followers through the optimal response to the MEC server strategy. Next, we define the utilities of MUs and the revenue of MEC servers, then model the computation offload problem as a Stackelberg game.

3.2 The Utility of MUs

The utility of MU i accomplishing type-j computation task is defined as

$$\begin{aligned} U_{i,j} = & \sum_{k=1}^{N} X_{\{\delta_{i,j}=k\}} \left(\alpha_{i,j} \left(E_{i,j}^{l} - E_{i,j}^{k} \right) - \varepsilon_k^e f_k^{e,i} \right) \\ & + \sum_{m=1}^{I} X_{\{\delta_{i,j}=-m\}} \left(\alpha_{i,j} \left(E_{i,j}^{l} - E_{i,j}^{m} \right) - \varepsilon_m f_i^m \right) \\ & + \sum_{i'=1, i' \neq i}^{I} X_{\left\{\delta_{i',j}=-i\right\}} \left(\varepsilon_i f_{i'}^{i} - \beta_i f_{i'}^{i} \right) \end{aligned} \tag{11}$$

Where $\alpha_{i,j}$ is the MU-specific parameter which demonstrates the sensitivity of MU i to the reduction in computation task execution energy consumption, $\alpha_{i,j} > 0$. ε_m and ε_k^e are the prices charged for the MU using the unit computation resource from the help user m and the MEC server k, respectively. β_m and β_k^e are severally corresponding cost. $f_k^{e,i}$ denotes the computation resources allocated by the MEC server k to MU i.

Since the MUs are rational, they maximize their utilities through making the offloading decision. In this case where the price set $\{\varepsilon_m, \varepsilon_k^e, m \in I, k \in N\}$ is given, the optimization problem for MU i can be formulated as

$$\begin{aligned}
&\max_{\{\delta_{i,j}, f_k^{e,i}\}} U_{i,j} \quad i \in I, j \in M, k \in N, m \in I \\
\texttt{s.t.}\ \ & C1: \max\left(\sum_{k=1}^{N} X_{\{\delta_{i,j}=k\}} T_{i,j}^k, T_{i,j}^l\right) \le T_{i,j}^{\max} \\
& C2: \max\left(\sum_{m=1}^{N} X_{\{\delta_{i,j}=-m\}} T_{i,j}^m, T_{i,j}^l\right) \le T_{i,j}^{\max} \\
& C3: \sum_{m=1}^{I} X_{\left\{\delta_{i,j}^m=-m\right\}} + \sum_{k=1}^{N} X_{\left\{\delta_{i,j}^k=k\right\}} \le 1 \\
& C4: X_{\{\delta_{i,j}=-m\}} = \{0,1\} \\
& C5: X_{\{\delta_{i,j}=k\}} = \{0,1\} \\
& C6: f_i^m \le f_m^{l,\max} \\
& C7: f_k^{e,i} \le f_k^{e,\max} + f_k^c
\end{aligned} \tag{12}$$

where f_k^c is the amount of computation resource bought from the network operator by MEC server k.

The offloading decision of the MU i not only depends on their own offloading demands, but also on the offloading strategies of other MUs, which makes Stackelberg game an appropriate approach to model the decision-making process. The players in this game are the MUs $\{N\}$. The strategy set for MU i can be expressed as $s_i = \{s_{i,-I}, ..., s_{i,-m}, ..., s_{i,1}, ..., s_{i,k}, ..., s_{i,N}\}$, where $s_{i,-m} = \delta_{i,j} f_i^m$, $s_{i,k} = \delta_{i,j} f_k^{e,i}$, $m \in I$, $k \in N$. Therefore, the player's strategic space can be denoted as $s = \{s_1 \times s_2 \times ... \times s_I\}$. Given s_{-i} as the offloading decisions for all MUs besides MU i, the utility of MU i can be given as $U_{i,j}(s_i, s_{-i})$, $i \in I$.

Theorem 1. *The computation task offloading game between the MUs is a concave multi-player game with Nash Equilibrium (NE).*

Proof. In the case where MU i offloads its computation task to help user m or MEC server k, as $f_i^m = [0, f_m^{i,max}]$, $f_k^{e,i} = [0, f_k^{e,max} + f_k^c]$. Thus, we have $s_{i,-m} = [0, f_m^{i,max}]$, $s_{i,k} = [0, f_k^{e,max} + f_k^c]$, $i \in I$, $m \in I$, $k \in N$. Replacing f_i^m, $f_k^{e,i}$ by $s_{i,-m}$, $s_{i,k}$ in (11), we can obtain $\frac{\partial^2 U_{i,j}}{\partial s_{i,-m}^2} = -\frac{\alpha_{i,j} P_{i,j} D_{i,j}}{s_{i,-m}^3} < 0$, $\frac{\partial^2 U_{i,j}}{\partial s_{i,k}^2} = -\frac{\alpha_{i,j} P_{i,j} D_{i,j}}{s_{i,k}^3} < 0$. For the given all MUs' offloading decisions besides MU i, the

MU i's payoff function $U_{i,j}(s_i, s_{(-i)})$ is strictly convex in terms of the variables $s_{i,-m}$, $s_{i,k}$, which holds for all MUs making offloading decisions. Therefore, the computation task offloading game between MUs is a strictly concave multi-player game, which exits a NE [10].

3.3 The Revenue of MEC Servers

As an offloading service provider, the MEC servers sells computation resources to MUs with the objective of making more profit. Since each of MUs can make any offloading decisions while meeting the specified computation task processing delay constraints, the MEC servers play the non-cooperative price decisions game with each other in order to determine the optimal computation resource price. Therefore, there will be competition between the MEC servers during the offloading process. In addition, when the MEC servers' own computation resources cannot meet the needs of the MUs, the computation resources can be purchased from the network operator. Since the total amount of available computation resources for the MEC servers affects the revenue of each MEC server, the amount of computation resources purchased from the network operator is also the decision of each MEC server in the offloading competition. Considering that competition is not perfect, each of MEC servers makes its offloading decision set (ε_k^e, f_k^c) based on its own available computation resources and the offloading demands, $k \in N$.

Since the computation resources from the network operator are often more expensive than those from the MEC server itself, we have $\varepsilon_k^c \succ \varepsilon_k^e$, $k \in N$. Therefore, each of MEC servers prefers to employing its own available computation resources first. Given the decision sets of other MEC servers, the revenue of employing decision (ε_k^e, f_k^c) for the MEC server k can be denoted as

$$\begin{aligned} &U_{mec}^k\left(\left(\varepsilon_k^e, f_k^c\right), \left(\varepsilon_{-k}^e, f_{-k}^c\right)\right) \\ &= \varepsilon_k^e \sum_{i=1}^{I} X_{\{\delta_{i,j}=k\}} f_k^{e,i} - \beta_k^e \min\left(\sum_{i=1}^{I} f_k^{e,i}, f_k^{e,\max}\right) - \varepsilon_k^c f_k^c \end{aligned} \tag{13}$$

Where $\left(\varepsilon_{-k}^e, f_{-k}^c\right)$ is the decision set of MEC servers besides server k. After that, the revenue optimization problem of the MEC server k can be expressed as

$$\begin{aligned} &\max_{\varepsilon_k^e, f_k^c} U_{mec}^k\left(\left(\varepsilon_k^e, f_k^c\right), \left(\varepsilon_{-k}^e, f_{-k}^c\right)\right) \\ \texttt{s.t.}\quad &C1: \varepsilon_k^c > \varepsilon_k^e \quad k \in N \\ &C2: f_k^c \geq 0 \quad k \in N \\ &C3: \beta_k^e > 0 \quad k \in N \end{aligned} \tag{14}$$

Lemma 1. *There exists an upper limit on the selling price of computation resources for each MEC server, i.e.,* $\varepsilon_k^e \leq \varepsilon_k^{e,\max}$, $k \in N$.

Proof. For specific computation tasks $b_{i,j}$, we have $U_{i,j} = \alpha_{i,j}\left(E^l_{i,j} - \frac{L_{i,j}}{R^k_{i,j}} - \frac{L_{i,j}D_{i,j}}{f^{e,i}_k}\right) - \varepsilon^e_k f^{e,i}_k$ according to (11) when MU i offloads computation task to the MEC server k. Since the MU i is rational, the MU i can offload computation tasks to the MEC server k only when $U_{i,j} > 0$. Let $Z_{i,j} = E^l_{i,j} - \frac{L_{i,j}}{R^k_{i,j}}$, we can get $\varepsilon^e_k < \frac{\alpha_{i,j}Z_{i,j}}{f^{e,i}_k} - \frac{D_{i,j}}{\left(f^{e,i}_k\right)^2}$. Then, let $Q\left(f^{e,i}_k\right) = \frac{\alpha_{i,j}Z_{i,j}}{f^{e,i}_k} - \frac{\alpha_{i,j}D_{i,j}}{f^{e,i}_k\left(\delta^M - \overline{\lambda_M}\right)}$. Nextly, we need to prove that $Q\left(f^{e,i}_k\right)$ has a maximum value in order to prove the existence of $\varepsilon^{e,\max}_k$. The first derivative of $Q\left(f^{e,i}_k\right)$ with regard to $f^{e,i}_k$ can be denoted by $\frac{\partial Q\left(f^{e,i}_k\right)}{\partial f^{e,i}_k} = -\frac{V}{\left(f^{e,i}_k\right)^2} + \frac{G\left(2f^{e,i}_k - S\right)}{\left(\left(f^{e,i}_k\right)^2 - Sf^{e,i}_k\right)^2}$, where $V = \alpha_{i,j}Z_{i,j}$, $G = \alpha_{i,j}L_{i,j}$, $S = P_{i,j}L_{i,j}$. Let $H = SV + G$, it can be seen that $\frac{\partial Q\left(f^{e,i}_k\right)}{\partial f^{e,i}_k} > 0$ when $\frac{H-\sqrt{GH}}{V} < f^{e,i}_k < \frac{H+\sqrt{GH}}{V}$. Therefore, the function $Q\left(f^{e,i}_k\right)$ has a maximum value $Q^{i,max}_k$. Taking into account that each of MEC servers is able to offload computation tasks from any MUs under the processing delay constraints, the sufficient condition for MEC servers that is capable of selling its computation resources to the MUs is $\varepsilon^e_k \le \varepsilon^{e,\max}_k = \max\left(Q^{1,\max}_k, Q^{2,\max}_k, ..., Q^{I,\max}_k\right)$, $k \in N$ (The same reason can be proved $\varepsilon_m \le \varepsilon^{\max}_m$, $m \in I$). Thus, there is an upper limit for the computation resource selling price of each MEC server.

Lemma 2. *There is an upper limit for the amount of computation resources purchased by each MEC server from network operator, i.e., $f^c_k \le f^{c,\max}_k$, $k \in N$.*

Proof. According to (13), if the resource requirements on the MEC server k are not exceed its supply capacity, i.e., $\sum_{i=1}^{I} X_{\{\delta_{i,j}=k\}} f^{e,i}_k \le f^{e,\max}_k$, the MEC server k should not purchase any resources from the network operator so as to maximize its revenue. Otherwise, (13) can be described as $U^k_{mec}\left(\left(\beta^e_k, f^c_k\right), \left(\beta^e_{-k}, f^c_{-k}\right)\right) = \varepsilon^e_k \sum_{i=1}^{I} X_{\{\delta_{i,j}=k\}} f^{e,i}_k - \beta^e_k f^{e,\max}_k - \varepsilon^c_k f^c_k$. $U^k_{mec}\left(\left(\beta^e_k, f^c_k\right), \left(\beta^e_{-k}, f^c_{-k}\right)\right) \ge 0$ due to that MEC servers are rational. Hence, we have $f^c_k \le \frac{\varepsilon^e_k \sum_{i=1}^{I} X_{\{\delta_{i,j}=k\}} f^{e,i}_k - \beta^e_k f^{e,\max}_k}{\varepsilon^c_k}$. Since Lemma 1 proves the maximum selling price $\varepsilon^{e,\max}_k$, we can get $f^c_k \le f^{c,\max}_k = \frac{\varepsilon^{e,\max}_k \sum_{i=1}^{I} X_{\{\delta_{i,j}=k\}} f^{e,i}_k - \beta^e_k f^{e,\max}_k}{\varepsilon^c_k}$.

Theorem 2. *There exits a Nash equilibrium in the game between the computation resource price determination and the network operator resource purchase among the MEC servers.*

Proof. For MEC server k, $k \in I$, its price strategy $\varepsilon^e_k \in (0, \varepsilon^{e,\max}_k]$ and the network operator purchase strategy $f^c_k \in (0, f^{c,\max}_k]$ according to Lemma 1 and Lemma 2. Therefore, the spaces of MEC servers' strategy sets $\{(\varepsilon^e_k, f^c_k), k \in N\}$

are convex, nonempty and compact. Moreover, from (13), it can seen that the revenue function of MEC server k is quasi-concave and continuous in terms of ε_k^e and f_k^c. Hence, the game has a Nash equilibrium [11,12].

3.4 Stackelberg Equilibrium

In the Stackelberg game of MUs computation task offloading process, the leaders are the MEC servers and the followers are the MUs. The MUs' equilibrium strategies in the game are the optimal response to the strategies announced by the MEC servers. According to (12), given strategy sets of the MEC servers, i.e., $\{(\varepsilon_k^e, f_k^c), k \in N\}$, the equilibrium strategy of MU i, i.e., $\left(\delta_{i,j}^*, f_k^{e,i*}\right)$, should meet the following condition:

$$U_{i,j}\left(s_{i,k}^*, s_{i,-k}^*\right) \geq U_{i,j}\left(s_{i,k}, s_{i,-k}^*\right) i \in I, j \in M \tag{15}$$

In like manner, the equalization strategies of the MEC server is based on the optimal strategies for the MUs' known response. The strategy set $(\varepsilon_k^{e*}, f_k^{c*})$ is an equilibrium strategy of the MEC server k, when

$$\begin{aligned} &U_{mec}^k\left(\left(\varepsilon_k^{e*}, f_k^{c*}\right), \left(\varepsilon_{-k}^{e*}, f_{-k}^{c*}\right), s_k\left(\varepsilon_k^{e*}, f_k^{c*}; \varepsilon_{-k}^{e*}, f_{-k}^{c*}\right)\right) \\ &\geq U_{mec}^k\left(\left(\varepsilon_k^e, f_k^c\right), \left(\varepsilon_{-k}^{e*}, f_{-k}^{c*}\right), s_k\left(\varepsilon_k^e, f_k^c; \varepsilon_{-k}^{e*}, f_{-k}^{c*}\right)\right) k \in N \end{aligned} \tag{16}$$

The game among MUs and the game among MEC servers have Nash Equilibrium according to Theorem 1 and Theorem 2, separately. Thus, there exists a Stackelberg Equilibrium in the Stackelberg game [13].

3.5 Proposed Algorithm

In order to solve the above computation offloading decision, we propose a distributed algorithm. In the algorithm, each of MEC servers begins by randomly choosing both the amount of computation resources purchased from network operator and its computation resource selling price. Taking into consideration that that each of MEC servers is rational, the price set by MEC server k is greater than or equal to the computation resource cost. Moreover, there exits an upper limit $\varepsilon_k^{e,\max}$ for the computation resource selling price of each MEC server. Therefore, the price of the randomly selected by the MEC server k should be in the interval $[\beta_k^e, \varepsilon_k^{e,\max}]$.

Theorem 3. *In the case where the MEC server k's selling price satisfies* $\beta_k^e \leq \varepsilon_k^e \leq \varepsilon_k^c$, *if the total computation resource demand from the MUs on the MEC server k is not less than its computation resource capacity upper limit* $f_k^{e,\max}$, *further lowering the selling price reduces the revenue of the MEC server k,* $k \in N$.

Algorithm 1. Computation offloading algorithm based on Stackelberg game (COAS)

Initialization:
a)The number of MUs I; the computation task, $b_{i,j}$, $i \in I, j \in M$
b)The computation resource of MU i and MEC server k: $f_i^{l,max}$, $f_k^{e,max}$, $k \in N$
c)The network operator computation resource selling price ε_k^c;
The MEC server k randomly selects its strategy set (ε_k^c, f_k^c);
Calculate the demand response of MUs;
Repeat
for MEC server k, $k \in N$ **do**
  if $\left(\sum_{i=1}^{I} X_{(\delta_{i,j}=k)} f_k^{e,i} \geq f_k^{e,\max}\right) \& (\varepsilon_k^e \leq \varepsilon_k^c)$ **then**
    Use (13) increase strategy to obtain the optimal price ε_k^{e*};
  else
    Get the optimal ε_k^{e*} by increasing or decreasing the strategies;
  end if
  Update the computation resource demands of the MUs;
  if $\left(\sum_{i=1}^{I} X_{(\delta_{i,j}=k)} f_k^{e,i} \leq f_k^{e,\max} \& f_k^c > 0\right) \|$
  $\left(\sum_{i=1}^{I} X_{(\delta_{i,j}=k)} f_k^{e,i} > f_k^{e,\max}\right) \&$
  $\left(\sum_{i=1}^{I} X_{(\delta_{i,j}=k)} f_k^{e,i} \neq f_k^{e,\max} + f_k^c\right)$ **then**
    $f_k^{c*} = \max\left(0, \sum_{i=1}^{I} X_{(\delta_{i,j}=k)} f_k^{e,i} - f_k^{e,\max}\right)$
  end if
end for
Until $\forall (\varepsilon_k^{e*}, f_k^{c*}) == (\varepsilon_k^e, f_k^c)$, $k \in N$.

According to the announced MEC servers' strategies, MUs are selected by a random order. By solving (12), each selected MU determines its offloading decision and the amount of computation resources that it needs to offload its computation task.

Based on the MUs' response, each MEC server first adjusts its computation resource selling price ε_k^e, $k \in N$. After the adjustment, if the demand for MEC server k still goes beyond its available computation resources, it determines to purchase more computation resources from the network operator. On the contrary, the MEC server k reduces f_k^c until f_k^c reaches 0. The MUs responds to the strategy changes of MEC servers. The MEC servers' strategies are iteratively updated until it is unchanged compared with the last iteration

In order to improve the efficiency of the proposed algorithm, some unrealistic strategies of the MEC server can be left from the iteration, as shown by the following theorem.

Proof. Since $\sum_{i=1}^{I} X_{\{\delta_{i,j}=k\}} f_k^{e,i} \geq f_k^{e,\max}$, according to (13), we can get $U_{mec}^k = \varepsilon_k^e X_{\{\delta_{i,j}=k\}} f_k^{e,i} - \beta_k^e f_k^{e,\max} - \varepsilon_k^c \left(\sum_{i=1}^{I} f_k^{e,i} f_k^{e,\max} \right)$. The revenue of VEC server k which cuts down computation resource price with the decrease θ is denoted by $U_{mec}^{k'}(\varepsilon_k^e - \theta)$. Taking into account that the each of MUs is rational, the price reduction has attracted more MUs to offload their computation tasks to the MEC server k. Let λ denote the increase in the demand for computation resources from the MUs. Then, we can denote $U_{mec}^{k'}(\varepsilon_k^e - \theta)$ as $U_{mec}^{k'}(\varepsilon_k^e - \theta) = (\varepsilon_k^e - \theta) \left(\sum_{i=1}^{I} X_{\{\delta_{i,j}=k\}} f_k^{e,i} + \lambda \right) - \varepsilon_k^e f_k^{e,\max} - \varepsilon_k^c \left(\sum_{i=1}^{I} X_{\{\delta_{i,j}=k\}} f_k^{e,i} + \lambda - f_k^{e,\max} \right)$. The difference between them can be given by $U_{mec}^{k'}(\varepsilon_k^e - \theta) - U_{mec}^k(\varepsilon_k^e) = (\varepsilon_k^e - \varepsilon_k^c)\lambda - \theta \left(\sum_{i=1}^{I} f_k^{e,i} + \lambda \right)$. As $\varepsilon_k^e \leq \varepsilon_k^c$, we get $U_{mec}^{k'}(\varepsilon_k^e - \theta) < U_{mec}^k(\varepsilon_k^e)$. Therefore, the decrease in the selling price ε_k^e reduces the server k's revenue.

The specific details of the algorithm we proposed are shown in Algorithm 1.

4 Simulation Results and Discussions

In this section, we evaluate the performance of CORA we proposed algorithm. We consider a 400 m * 400 m area, where SBSs and MUs are randomly distributed and the radius of each SBS is set as 100 m. Besides, the channel gain is denoted by d^{α}, where d is the distance between the SBS and the MU, α represents the path loss exponent, of which value is -3. Moreover, other main parameter settings are shown in Table 1 [14,15].

Table 1. The simulation parameters.

Simulation parameters	Value
The bandwidth of each Opportunity network channel B^m	20 MHz
The bandwidth of each Cellular network channel	40 MHz
The transmission power of the MU $P_{i,j}$	200 mW
The type of computation task M	5
The maximum delay of the computation task $T_{i,j}^{\max}$	[0.2, 1.5] s
The computation capacity of the MU $f_i^{l,\max}$	[1, 2] GHz
The computation capacity of the MEC server $f_k^{e,\max}$	[8, 10] GHz

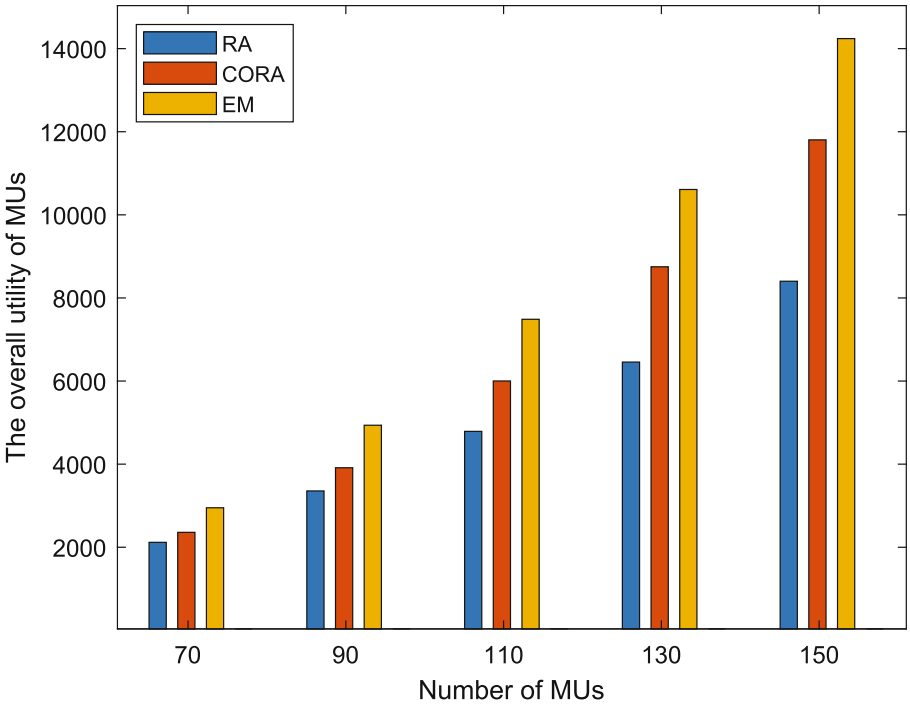

Fig. 2. The comparison of MUs utility among CORA, RA and EM

To prove the effectiveness of the CORA proposed solution, we compare its performance with the other two baseline algorithms as shown in Fig. 2, where EN and RA denotes Enumeration method and random computation offloading scheme, respectively. The MU utility of the CORA algorithm is very close to EN, which should theoretically be the optimal algorithm. However, it should not be overlooked that EN has very high time complexity compared with the CORA. The MUs' utility of the RA scheme is sometimes higher than EN, sometimes otherwise, which is because of its randomness. Therefore, CORA can be considered as the optimal algorithm in terms of both the time complexity and the utility of the MUs.

To further demonstrate the performance of the CORA algorithm, we compare it to two other computation offloading strategies: Proportional Price and Dynamic Price. In Fig. 3, the computation offloading system with the proportional price strategy has the lowest benefit. Since the price in this case is directly proportional to the computation resource cost of the MEC server, it cannot be dynamically adjusted according to the supply and demand situation. That's why these two options get more profit than the proportional price strategy.

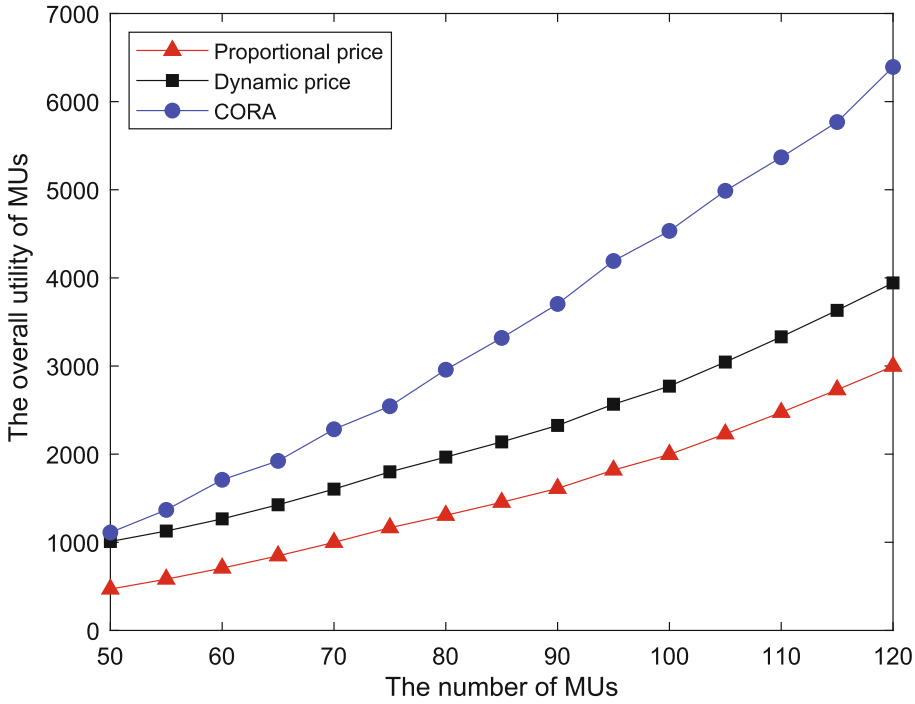

Fig. 3. The overall utility of the MUs with different offloading schemes.

Whereas, due to the limitation of the MEC servers' computation resource capacity, the revenue obtained by the price scheme without BCS cannot be increased with the increase of the number of MUs. In our proposed CORA algorithm, when BCS is used to compensate for insufficient computation resources, the price can be optimally adjusted. Therefore, compared with the other two offloading strategies, the CORA we proposed can gain the highest utility, whether it is a small amount or a large number of MUs.

Figure 4 illustrates the impact of BCS computation resource prices on MEC servers' revenue as the number of MUs increases. It can be seen that all incomes in these cases decrease when the computation resource price rises. Due to the limitation of the MEC server's computation resource, more computation tasks is capable of being offloaded to the BCS as the number of MUs increases. Consequently, the revenue with more MUs is more susceptible to the BCS computation resource and is more likely to decrease as prices increase. Besides, we can find that when the computation price is 0.9, the revenue of 200 MUs is nearly equal to the case where the number of MUs is 160. The reason is that there exits an

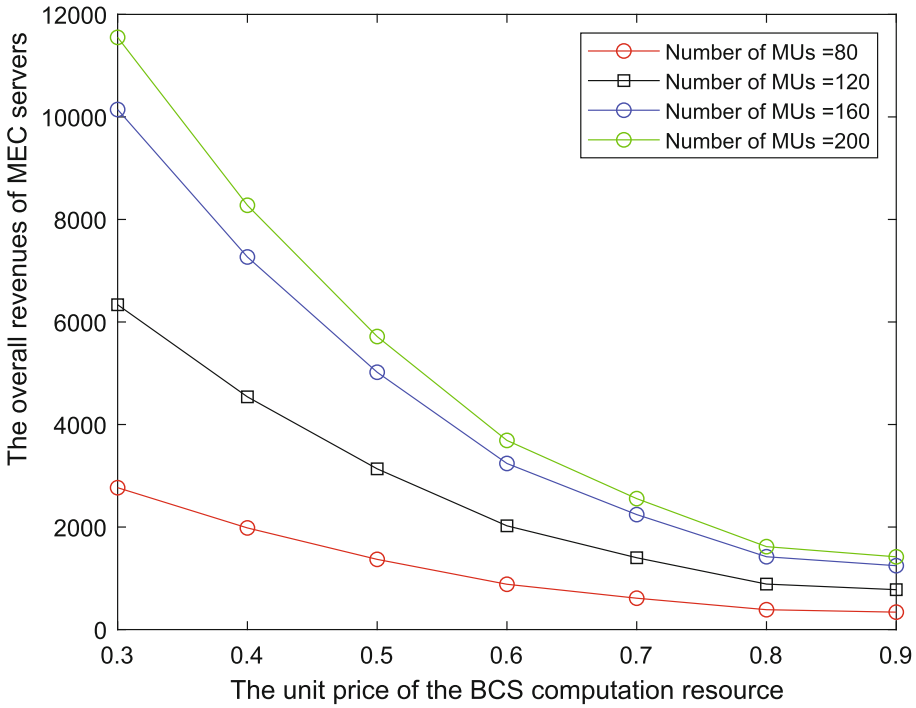

Fig. 4. The overall revenues of the MEC servers affected by BCS resource price.

upper limit for MEC servers' computation resource price according to Lemma 1. The MEC servers may not buy computation resource from the BCS when its price becomes high. In addition, due to computation resource constraints, the number of offloading services for MEC servers is limited. Therefore, in the case of a high BCS price and a large number of vehicles, the benefits obtained by the MEC servers are almost the same.

Figure 5 shows convergence of the proposed algorithm in terms of MUs' utility and is a perfect proof that the system can reach the NE, which can further explain that CORA cannot only get close to the optimal result by the lower time complexity than EM but also it is stable.

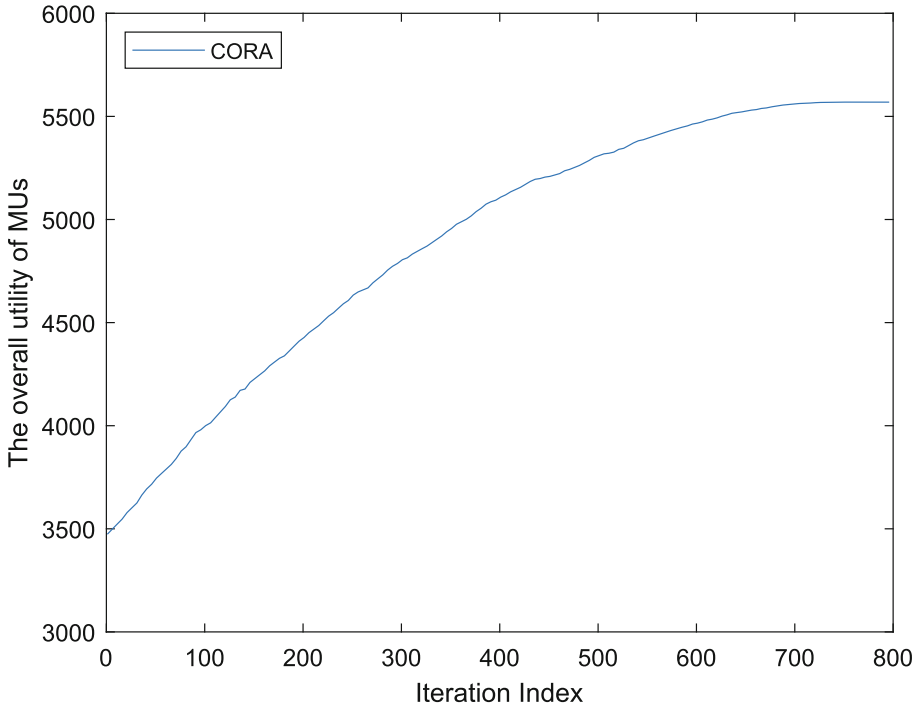

Fig. 5. Convergence of the proposed algorithm

5 Conclusion

In this paper, we design a multi-level computation offloading strategy and consider the heterogeneity of computation tasks and computation resource backup pool while making offloading decision in multi-access networks. Firstly, the computation offloading model is introduced for each computation strategy. Then the main problem is formulated as a Stackelberg game problem, and a global optimal algorithm based on game theory with lower time complexity compared with EM and RA is proposed and the numerical simulation results show the efficiency and stability of the proposed algorithm. For future work, in order to improve the performance of the MEC system, we will consider a resource s allocation and design a more efficient algorithm.

Acknowledgment. This work has been supported by STATE GRID Corporation of China science and technology project" Research and application on key technologies of power wireless heterogeneous network convergence" (5700-201919236A-0-0-00).

References

1. Kan, T., Chiang, Y., Wei, H.: QoS-aware mobile edge computing system: multi-server multi-user scenario. In: 2018 IEEE Globecom Workshops (GC Wkshps), Abu Dhabi, United Arab Emirates, pp. 1–6 (2018)
2. Wang, S., Zhang, X., Zhang, Y., Wang, L., Yang, J., Wang, W.: A survey on mobile edge networks: convergence of computing, caching and communications. IEEE Access **5**, 6757–6779 (2017)
3. Cao, X., Wang, F., Xu, J., Zhang, R., Cui, S.: Joint computation and communication cooperation for energy-efficient mobile edge computing. IEEE Internet Things J. **6**(3), 4188–4200 (2019)
4. Dai, Y., Sheng, M., Liu, J., Cheng, N., Shen, X.: Resource allocation for low-latency mobile edge computation offloading in NOMA networks. In: 2018 IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, United Arab Emirates, pp. 1–6 (2018)
5. Chen, M., Liang, B., Dong, M.: Joint offloading and resource allocation for computation and communication in mobile cloud with computing access point. In: IEEE INFOCOM 2017 - IEEE Conference on Computer Communications, Atlanta, GA, pp. 1–9 (2017)
6. Guo, F., Zhang, H., Ji, H., Li, X., Leung, V.C.M.: Energy efficient computation offloading for multi-access MEC enabled small cell networks. In: 2018 IEEE International Conference on Communications Workshops (ICC Workshops), Kansas City, MO, pp. 1–6 (2018)
7. Yang, Y., Wang, K., Zhang, G., Chen, X., Luo, X., Zhou, M.: Maximal energy efficient task scheduling for homogeneous fog networks. In: IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Honolulu, HI, pp. 274–279 (2018)
8. Tran, T.X., Pompili, D.: Joint task offloading and resource allocation for multi-server mobile-edge computing networks. IEEE Trans. Veh. Technol. **68**(1), 856–868 (2019)
9. Marin, R., Ciobanu, R., Dobre, C.: Improving opportunistic networks by leveraging device-to-device communication. IEEE Commun. Mag. **55**(11), 86–91 (2017)
10. Rosen, J.B.: Existence and uniqueness of equilibrium points for concave N-person games. Econometrica **33**(3), 520–534 (1965)
11. Debreu, G.: A social equilibrium existence theorem. Proc. Natl. Acad. Sci. U.S.A. **38**(10), 886–893 (1952)
12. Zhang, K., Mao, Y., Leng, S., Maharjan, S., Zhang, Y.: Optimal delay constrained offloading for vehicular edge computing networks. In: 2017 IEEE International Conference on Communications (ICC), Paris, pp. 1–6 (2017)
13. Maharjan, S., Zhu, Q., Zhang, Y., Gjessing, S., Basar, T.: Dependable demand response management in the smart grid: a stackelberg game approach. IEEE Trans. Smart Grid **4**(1), 120–132 (2013)
14. Guo, F., Ma, L., Zhang, H., Ji, H., Li, X.: Joint load management and resource allocation in the energy harvesting powered small cell networks with mobile edge computing. In: IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Honolulu, HI, pp. 299–304 (2018)
15. Guo, H., Zhang, J., Liu, J., Zhang, H., Sun, W.: Energy-efficient task offloading and transmit power allocation for ultra-dense edge computing. In: 2018 IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, United Arab Emirates, pp. 1–6 (2018)