International Journal of
Distributed
Sensor Networks

International Journal of Distributed Sensor Networks
2019, Vol. 15(7)

DOI: 10.1177/1550147719861556
journals.sagepub.com/home/dsn
SAGE

Mobile edge computing resource allocation: A joint Stackelberg game and matching strategy

Shaoyong Guo[1], Xing Hu[1], Gangsong Dong[2], Wencui Li[2] and Xuesong Qiu[1]

Abstract

Mobile edge computing has attracted great interests in the popularity of fifth-generation (5G) networks and Internet of Things. It aims to supply low-latency and high-interaction services for delay-sensitive applications. Utilizing mobile edge computing with Smart Home, which is one of the most important fields of Internet of Things, is a method to satisfy users' demand for higher computing power and storage capacity. However, due to limited computing resource, how to improve efficiency of resource allocation is a challenge. In this article, we propose a hierarchical architecture in Smart Home with mobile edge computing, providing low-latency services and promoting edge process for smart devices. Based on that, a Stackelberg Game is designed in order to allocate computing resource to devices efficiently. Then, one-to-many matching is established to handle resource allocation problems. It is proved that the allocation strategy can optimize the utility of mobile edge computing server and improve allocating efficiency. Simulation results show the effectiveness of the proposed strategy compared with schemes based on auction game, and present performance with different changing system parameters.

Keywords

Mobile edge computing, Smart Home, Stackelberg game, one-to-many matching, computing resource allocation

Date received: 27 February 2019; accepted: 13 May 2019

Handling Editor: Feng Ye

Introduction

Imminent fifth-generation (5G) cellular networks are expected to play a major role in supporting Internet of Things (IoT) to achieve ubiquitous coverage, inherent security, high connectivity, and quality of service (QoS). It can provide unique advantages considering its widespread infrastructures already deployed on a large scale, and service reliability offered by a mature wireless standard with various technologies. Due to the limited existing spectrum and the need for high-speed data rates, 5G is expected to incorporate higher frequency spectra in the millimeter wavebands, such as 30 and 60 GHz.[1,2] However, with rapid extension, the IoT need computation and storage in proximity of sensors and actuators to support delay-sensitive and computation-intensive applications. According to Cisco Global Cloud Index Networking,[3] data generated by various end devices are expected to reach 600 ZB (zettabyte) in 2020. Traditional IoT

[1]State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China
[2]State Grid Information & Telecommunication Technology Group Co., Ltd, State Grid Henan Electric Power Company, Henan, China

Corresponding author:
Xuesong Qiu, State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China.
Email: xsqiu@bupt.edu.cn

frameworks adopt cloud computing to deal with large torrent of data generated by all kinds of IoT devices. Generally speaking, large-scale data centers or cloud is built in remote areas far from edge terminals. These architectures perform poorly in real-time response, edge security, and edge processing, which is intolerable for IoT applications that require real-time response. Under this circumstance, mobile edge computing (MEC) is proposed as a critical solution to provide low-latency, fast-response, and high-interaction service for edge terminals.

As a new computing paradigm that combines Internet, computation, and storage at the edge of network, MEC has promising benefits.[4] With multiple edge nodes, called access points (APs), deployed in the edge of network, MEC brings "cloud" to the edge. It can meet real-time requirements or achieve interactive behaviors of a large variety of new edge applications, as it is deployed much closer to edge terminals. Furthermore, MEC also relieves data congestion within core Internet and satisfies demands of devices for a higher computational power and storage capacity, allowing them to offload their tasks. It has been applied in many aspects of IoT. Hui et al.[5] proposed a fog computing-based resolution framework to efficiently resolve the individuals identity. Su et al.[6] developed an edge caching scheme in vehicular networks based on edge computing. Tang et al.[7] designed a hierarchical distributed fog computing architecture to support data-intensive analysis in future smart cities.

Nevertheless, extending of MEC in Smart Home meets challenges caused by limited computation and bandwidth resources. Research suggests that Chinese market of Smart Home is expected to increase by 48% on average each year. With the demand for Smart Home increasing rapidly, it faces severe deficiency of bandwidth and wasting of computational resources. Definitely, it is established at home for a better life quality, combining technology and services through networking process.[8,9] Applications in the system can communicate with one another and provide instant services in different phenomenon, raising high standard for computational capability and bandwidth. Faced with these issues, improving resource allocation efficiency is inevitable in Smart Home. Therefore, this article discusses how to design effective resource allocation strategy based on MEC. To meet demands for low-latency services of delay-sensitive applications, APs are deployed near the Smart Home system, forming a three-layered architecture. Applications can utilize computational and storage resources of AP to conduct tasks. In addition, effective allocation strategy needs to be designed, aiming to satisfy requirements of more applications with limited resources.

In this article, we propose a strategy for computational resource allocation to improve performance of Smart Home system with MEC. A hierarchical architecture is designed, consisting of user equipment (UE), AP, and MEC server. AP can aggregate data generated by UE and improve edge processing. Then, based on Stackelberg game and matching, an efficient and dynamic allocation strategy for computational resources is put up. Our main contributions are summarized as follows:

1. Build a resource allocation model for Smart Home system with MEC. A three-tier framework is built to support applications' demands for real-time interaction services. And optimization problems are formulated, limited by computational resources, taking diverse tolerance in delay and rate of applications into consideration.
2. A Stackelberg Game is proposed to represent relations between AP and UE. In Stackelberg Game, AP sets prices for virtual computational resource unit as a leader, and UE can determine the number of requiring resource as a follower. Through the game, higher utilities for UE and AP can be obtained.
3. One-to-many matching is introduced to maximize utility of MEC server. AP competes for computational resources owned by MEC server, forming a one-to-many matching. Similarly, AP reallocates their resources to users. It is proved that MEC server can obtain the best outcomes according to matching. In other words, computational resources can be distributed optimally.
4. The performance of the proposed system is evaluated and compared with the existing model with respect to performance metrics.

The rest of this article is organized as follows: The "Related work" section reviews the related work. The "System model" section presents the system model, including system architecture, network model, and problem analysis. An resource allocation strategy based on the Stackelberg game and one-to-many matching is introduced in the "Resource allocation strategy based on game theory and one-to-many matching" section, after which we propose the experimental results and corresponding discussions in the "Simulation results and analysis" section. The "Conclusion and future work" section is the conclusion and future work.

Related work

Cloud computing in IoT

During the last decade, cloud computing drew considerable attention as it offers lower cost solutions for fluctuating and unforeseen computational demand.[10] It has been adopted by many organizations or enterprises to meet their demands for massive data processing.

In terms of environment monitoring, in order to derive the physical topology at the cloud for effective real-time event detection, Yu et al.[11] proposed a cloud-orchestrated physical topology discovery scheme for large-scale IoT systems. A network-wide three-dimensional (3D) localization algorithm is developed to improve the efficiency and accuracy of topology discovery. In order to establish a sensing as a service model, Perera et al.[12] proposed a scalable energy-efficient data analytics platform for on-demand distributed mobile crowd sensing called C-MOSDEN, which considered different domains to facilitate all conditions.

In manufacturing, Tao et al.[13] gave a brief overview of the application of cloud computing in IoT first. And then they established a cloud computing and IoT-based cloud manufacturing service architecture and designed technology systems, to realize the full sharing, free circulation, and optimal allocation of manufacturing resources and capabilities. In IoT big data storage, Cai et al.[14] gave an overview about current technology and research in IoT application, and analyzed perspectives and challenges in establishing IoT-based data storage systems in cloud computing.

Though conventional data center design is highly regular with identical servers and networking hardware aligned in a grid-like fashion, the centralized form of remote computing resources is not compatible with enormous traffic originated from geographically distributed edge devices. In microgrids, Harmon et al.[15] presented a cloud-based and hybrid wireless mesh communication framework in microgrids. A diffusion-based, fully distributed algorithm and a quasi-distributed approach on wide-area Internet-based cloud are implemented. To overcome security challenges for data mutuality between two parties in IoT, Wang et al.[16] presented a framework based on cryptographic methods to support data security in cloud-assisted IoT. However, other types of security issues, such as authentication, were not considered in this article.

Above all, meeting the needs and QoS requirements of various tasks that are demanded by IoT devices cannot be achieved by solely employing cloud servers or edge computing servers. In order to cover and serve wide range of envisioned services, both cloud and edge computing facilities need to be incorporated into a joint architecture.

Resource allocation strategy based on edge computing in IoT

Recently, many literatures have studied resource allocation strategies in different edge computing scenes. Wang et al.[17] proposed an integrated framework form computation offloading and interference management in wireless cellular networks with MEC. They performed resource allocation using the graph coloring methods. Nevertheless, they did not consider competence between applications. Munoz et al.[18] focused on wireless application offloading for radio and computational resources, optimizing trade-off between energy consumption and latency in femto-cloud. The allocation framework could save terminal battery and reduce latency in executing application.

Game theory is adopted by many works to solve data offloading. Zhang et al.[19] presented a coalitional game based on pricing scheme to offload big data to the MEC server. Lian et al.[20] designed a game-theoretic framework under the communication cost constraint to enable economically viable communications. They optimized performance of MEC depending on location of edge servers. Zhang et al.[21] gave an algorithm based on differential game to control energy consumption for cognitive radio (CR) system. Zhang et al.[22] discussed a joint optimization approach combining Stackelberg Game and matching to achieve computing resource allocation in three-tier networks. It realizes utility maximization for servers based on fog computing.

Although these works have dedicated to different edge computing scenes, most of them focused on power control or resource allocation for servers or base stations. In Smart Home, considering requirement for high QoS and real-time response of delay-sensitive applications, it is crucial to establish an efficient and distributed framework with low interference based on MEC.

System model

System architecture

To meet requirements of different applications for real-time interaction and low-latency services in Smart Home, a three-tier architecture is considered, where UE can offload its service to MEC data centers through an AP as shown in Figure 1. APs are located in middle layer, linking MEC server in upper layer and serving UE in bottom layer. Generally, UE is connected wirelessly in Smart Home. It consists of different subsystems, such as lighting control, surveillance, and HVAC (heating, ventilation, and air conditioning),[23] which supply different services and have diverse tolerance for delay and transmission speed. Therefore, an allocation strategy for computational resource based on pricing mechanism is designed, aiming to meet demands of applications with higher computational speed or lower latency for priority.

Assume that the computational resource of MEC server is stable during a certain time. The server can allocate resources to linked wireless base stations. Each base station has an AP that is associated with covered

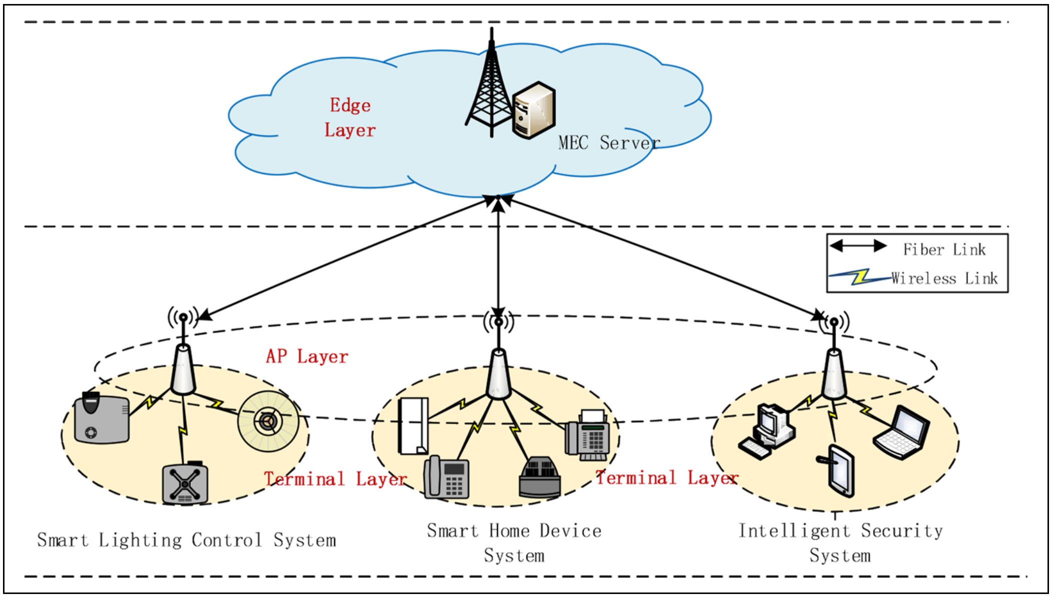

Figure 1. System architecture.

UE. When executing low-latency or real-time interaction tasks, UE can submit requests for AP and offload tasks to MEC server through telecommunication links after receiving response from AP.

In this article, MEC server links with AP directly and distributes resources to them. After AP purchases computational resources from the server, it can make optimal allocating decisions on connected UE to gain better utility.

Network model

It is assumed that there are K APs, labeled as $A = \{a_1, a_2, \ldots, a_K\}$. The number of UE that is connected to a_k is M_k, denoted as $S^k = \{s_1^k, s_2^k, \ldots, s_{M_k}^k\}$. The workload arrival rate of user s_j is denoted by λ_j. Then, MEC server has a normalized preference list over all AP, denoted as $L = \{L_1, L_2, \ldots, L_K\}$. Similarly, a_k has a preference list over its users, represented as $N^k = \{\eta_1^k, \eta_2^k, \ldots, \eta_{M_k}^k\}$. Define unit amount of computing resources that can be distributed by MEC server as "computing resource block (CRB)." Each CRB can provide computing service at a rate of μ. Parameters used in this article are listed in Table 1.

User terminals can get CRBs from MEC server to conduct high QoS applications. The quality of QoS can be measured by service delay t_j. It consists of queuing delay o_j at servers and network delay h_j. Then, t_j for s_j can be calculated by

$$t_j = h_j + o_j \tag{1}$$

According to models in Liu et al.,[24] o_j can be presented by

$$o_j = \frac{\lambda_j}{\mu - \frac{\lambda_j}{q_j}} \tag{2}$$

Table 1. The table of parameters.

Notation	Meaning
K	Amounts of AP
a_k	The kth AP in the system
M_k	Amounts of UE connected to a_k
M	Amounts of CRBs
A	Set of AP
S^k	Set of UE that connected to a_k
μ	Service rate of each CRB
s_j^k	The jth UE connected to a_k
λ_j	The workload arrival rate of s_j^k
t_j	The service delay of s_j^k
o_j	The queuing delay in servers
h_j	The network delay
q_j	CRB amounts that UE requires
l_j^k	Distance from s_j^k to the server
$W_{k_j}^s$	Utility of s_j^k
W_k^a	Utility of a_k
W^o	Utility of the MEC server
L	The preference list of MEC server over all AP
N^k	The preference list of a_k over connected UE
$\alpha_j^k, \beta_j^k, \gamma_j^k$	Weight factors in the utility function of UE
r_j^k	Setting price of AP over UE
p_k	Renting price that AP rents CRB from servers
c_k	Transmission cost from MEC server to AP
Q_k	CRB amounts that AP requires
$\tilde{Q}_k$	CRB amounts that AP is allocated
t_{th}	The upper bound of delay tolerance

AP, access point; UE, user equipment; CRB, computing resource block; MEC, mobile edge computing.

Research suggests that the network delay h_j is influenced by many unpredictable factors such as transmission distance, traffic flow, and hardware equipment. It is recognized that h_j follows a linear function with the total distance, l_j, which equals to the distance from server to AP plus length from AP to user. Set scalar as θ, and h_j can be represented as follows

$$h_j = \theta \cdot l_j \tag{3}$$

Users have different valuation of the edge computing services, depended on certain factors such as latency and service rates. These factors influence the computing resource demands of users. The demand should be derived to maximize the utility of terminals. Therefore, a price mechanism is introduced in the article. UE can gain reward in the process of executing real-time tasks. The number of tasks is related to the workload arrival rate, λ_j. Meanwhile, it needs to undertake payments of purchasing CRBs. Besides, s_j^k ought to consider cost for service delay. It is worth noting that there is an upper bound for the service delay. If the delay is larger than the threshold, the task would be regarded as a failure. Given weight factors $\alpha_j^k, \beta_j^k, \gamma_j^k$, utility function of s_j^k can be concluded

$$W_{k_j}^s = \alpha_j^k \times \lambda_j^k - \beta_j^k \times q_j^k \times r_j^k - \gamma_j^k \times t_j^k \tag{4}$$

According to requests from UE, AP can receive revenues for distributing CRBs. And it should pay for MEC server for renting CRBs as well. CRB amounts that AP allocates to connected UE should not exceed what it rents from MEC server. Therefore, utility function of a_k through incomes minus losses can be calculated

$$W_k^a = \left(\sum_{j=1}^{M_k} r_j^k \times q_j^k \right) - p_k \times Q_k \tag{5}$$

MEC server can get payments from APs for renting CRBs. The expenditure of MEC server includes costs in transmitting and maintaining normal function of the network. To sum up, discounted utility function for MEC server is

$$W^o = \sum_{i=1}^{K} l_i \times (p_i - c_i) \times Q_i \tag{6}$$

Through adopting the price mechanism, computing tasks and resources can be measured by incomes or expenditure. In this way, utility of these objects can represent the efficiency of computational resource allocation.

Problem analysis

In summary, problems that need to be considered are listed as follows.

1. *Amount of CRBs that UE purchases from AP*. In the system, UE requires to purchase CRBs from APs. Since the CRB amounts that UE demands is closely influenced by the upper bound of service delay and setting price of AP, UE should determine the optimal amounts of CRB for higher utility.
2. *Pricing problem for AP*. AP needs to rent CRBs from MEC server and sell them to connected UE. To obtain maximum outcomes, AP intends to sell with high setting prices and pay for low renting prices. However, there is a pair of contradictions: facing high setting prices, UE will reduce CRB amounts purchased from AP at the cost of larger latency. In an extreme situation, UE will give up if it cannot obtain profits from conducting tasks. Meanwhile, the lower renting price AP pays, the smaller CRBs that AP can be allocated from MEC server. Therefore, predicting the reactions of UE, AP is required to balance service price and renting price so as to gain the maximum revenues.
3. *Allocation problem between MEC server and AP*. According to different renting price and distance in multi-AP scenario, MEC server obtains various profits from providing CRBs to AP, causing different preference over all AP. The server aims to reach a maximum utility based on the preference list over all AP.
4. *Allocation problem between AP and UE*. After matching between MEC server and AP, the server has distributed its CRBs to all AP. If CRBs that AP rents from server are less than that it requires, AP needs to determine allocation strategy over UE. On the basis of serving price, AP allocates resource on all UE to achieve a higher utility.

Resource allocation strategy based on game theory and one-to-many matching

In this section, optimal problems and allocation strategy based on game theory and matching are proposed. Utility functions of MEC server, AP, and UE are formulated as optimization problems according to equations (4)–(6), respectively. To obtain maximum utility for MEC server, first, a Stackelberg Game is established to find optimal CRB number for UE and solve pricing problem for AP. Second, CRBs are distributed through a one-to-many matching between MEC server and AP. Finally, an optimal allocation scheme between AP and UE is given.

Optimization problems

Taking workloads arrival rate, service delay, and number of requiring resources into accounts, utilities of

MEC server, AP, and UE are formulated. In this way, they can be used to measure efficiency and quantity of supplying real-time services. In this section, utilities of three objects are formulated as optimization problems.

t_{th} represents maximum tolerance of service delay for s_j^k. Queuing delay cannot surpass t_{th}, or it is a failure task for users. Utility of UE should be greater than zero. When serving price for s_j^k is determined, CRB number that AP requires can be calculated. An optimization problem for s_j^k is formulated as follows

$$\max_{q_j^k} \ W_{k_j}^s\left(q_j^k | r_j^k\right)$$
$$\text{subject to} \begin{cases} t_j^k \leqslant t_{th} \\ \alpha_j^k \times \lambda_j^k \geqslant \beta_j^k \times q_j^k \times r_j^k + \gamma_j^k \times t_j^k \\ q_{\mathrm{j}}^k \geqslant 0 \end{cases} \tag{7}$$

After observing behaviors of UE, AP can set a renting price for purchasing CRBs from MEC server. It may cause interference when CRB amount that AP allocates to UE exceeds what AP owns, since same bandwidth frequency may be shared by different users. Utility of AP is also greater than zero. Optimal CRB amount required by all AP is denoted by Q_k^*. Prices determined by other devices except for s_j^k are represented by $\rightharpoonup r_{-j}^k$. In order to gain a maximum utility at a specific serving price for s_j^k, optimization problem of AP can be formulated as below

$$\max_{r_j^k} \ W_k^a\left(r_j^k | Q_k^*, p_k, \ \rightharpoonup r_{-j}^k\right)$$
$$\text{subject to} \begin{cases} \sum_{j=1}^{M_k} \left(r_j^k \cdot q_j^k\right) \geqslant p_k \cdot Q_k \\ r_k \geqslant 0 \\ \sum_{j=1}^{M_k} q_j^k \leqslant Q_k \\ Q_k \geqslant 0 \end{cases} \tag{8}$$

When MEC server receives renting prices from AP, it allocates CRBs relying on the preference list over AP. Renting price set by AP should be greater than transmission cost that MEC server undertakes. As total number of CRBs in network is M, available number that can be rented by AP must be less than M. Q_{-k}^* and p_{-k}^* represent optimal CRB number and optimal renting price of other AP except a_k, respectively. The optimization problem of MEC server is

$$\max_{Q_k} \ W^o\left(Q_k | Q_{-k}^*, p_{-k}^*\right)$$
$$\text{subject to} \begin{cases} p_k \geqslant c_k \\ \sum_{i=1}^{K} Q_i \leqslant M \end{cases} \tag{9}$$

Stackelberg game for AP and UE

In Stackelberg game,[25] AP acts as a leader, predicting serving prices and required CRB number for user devices. UE acts as a follower, purchasing CRBs from AP when knowing the setting price. In the whole process, AP can observe actions of UE and determine service price finally. According to the Stackelberg game, lemma 1 can be concluded as follows:

Lemma 1. *When serving price that AP sets is r_j^k, the optimal amount of CRBs to reach maximum utility for user s_j^k is*

$$q_j^{k^*} = \frac{\lambda_j^k}{\mu} + \frac{\lambda_j^k}{\mu\sqrt{\frac{\beta_j^k \times r_j^k}{\gamma_j^k}}} \tag{10}$$

Proof. On the basis of utility function of s_j^k, the second derivation of $W_{k_j}^s$ with respect to q_j^k is

$$\frac{\partial^2 W_{k_j}^s}{\partial (q_j^k)^2} = \frac{-2\mu \times \gamma_j^k \times \left(\lambda_j^k\right)^2}{\left(\mu \times q_j^k - \lambda_j^k\right)^2} \tag{11}$$

For $\partial^2 W_{k_j}^s / \partial (q_j^k)^2 < 0$, the first derivation of $W_{k_j}^s$ follows a decreasing relationship with respect to q_j^k. When $\partial W_{k_j}^s / \partial q_j^k = 0$, value of q_j^{k*} is optimal number of CRBs for UE to gain a maximum utility. The first derivation of $W_{k_j}^s$ is

$$\frac{\partial W_{k_j}^s}{\partial q_j^k} = -\beta_j^k \times r_j^k + \frac{\gamma_j^k \times \left(\lambda_j^k\right)^2}{\left(\mu \times q_j^k - \lambda_j^k\right)^2} \tag{12}$$

Let the first derivation of W_j^k with respect to q_j^k equal to zero, and we can calculate an optimal number of CRBs purchased by UE as shown in lemma 1.

According to equation (7) and equation (11), the optimization problem of AP can be adjusted as follows

$$\max_{r_k} \ W_k^{\mathrm{a}}\left(r_j^k | Q_k^*, p_k, \ \rightharpoonup r_{-k}\right)$$
$$\text{subject to} \begin{cases} \sum_{i=1}^{M_k} \left(\frac{\lambda_j^k}{\mu} + \frac{\lambda_j^k}{\mu \times \sqrt{\frac{\beta_j^k \times r_j^k}{\gamma_j^k}}}\right) \times r_j^k \geqslant p_k \times Q_k \\ r_j^k \geqslant 0 \\ \sum_{i=1}^{M_k} q_i^k \leqslant Q_k \\ Q_k \geqslant 0 \end{cases} \tag{13}$$

Utility function of a_k can also be updated

$$W_k^a = \sum_{i=1}^{M_k}\left(\frac{\lambda_j^k}{\mu} + \frac{\lambda_j^k}{\mu \times \sqrt{\frac{\beta_j^k \cdot r_j^k}{\gamma_j^k}}} \times r_j^k - p_k \times Q_k\right) \quad (14)$$

First derivation of W_k^a is formulated as below

$$\frac{\partial W_k^a}{\partial r_j^k} = \frac{\lambda_j^k}{\mu} + \frac{\lambda_j^k}{2\mu \times \sqrt{\frac{\beta_j^k \times r_j^k}{\gamma_j^k}}} \quad (15)$$

As $\frac{\partial W_k^a}{\partial r_j^k} > 0$, utility of AP is positive correlated with r_j^k.

Furthermore, service delay of UE cannot surpass maximum tolerance of delay t_{th}. A low bound of t_{th} according to equation (2) can be inferred as follows

$$q_j^k \geqslant \frac{\lambda_j^k \times t_{th}}{\mu \times t_{th} - \lambda_j^k} \quad (16)$$

Therefore, following results in equation (10), an upper bound of r_j^k can be calculated, which is just the optimal price to obtain maximum utility of a_k according to equation (15)

$$r_j^k = \frac{\gamma_j^k}{\beta_j^k} \times \left(\frac{\mu \times t_{th} - \lambda_j^k}{\lambda_j^k}\right)^2 \quad (17)$$

According to equation (10), AP can gain the optimal number of CRBs that it expects to rent from MEC server as below

$$\tilde{Q}_k = \sum_{i=1}^{M_k} q_i^k \quad (18)$$

One-to-many matching for MEC server and AP

Knowing the number of CRBs that AP wants to rent, MEC server makes decision on how to distribute CRBs to AP. Based on renting price and transmission cost, server has different preference over all AP, denoted as $L = \{L_1, L_2, \ldots, L_K\}$. AP with a higher preference has priority in the process of allocating computing resource. Different distances between MEC server and AP affect communication cost among them. And utility of MEC server is related to the cost and renting price. Therefore, the value of preference over AP can be measured by communication cost and renting prices

$$L_i = p_i - c_i \quad (19)$$

According to analysis of formulated framework, a one-to-many matching between MEC server and AP is designed. As shown in Algorithm 1, after constructing a preference list over all AP in step 1, one pointer is set as an indicator pointing at the most preferred AP initially in step 2. In steps 8–11, if server can supply more CRBs than what AP needs, demands of AP can be satisfied. Otherwise, MEC server will allocate all the left CRBs to AP. At the end of each round, if all CRBs of server have been allocated, the pointer will remain unchanged, or it will move to next AP. In summary, AP with a low preference may gain insufficient CRBs. Then, it needs to reallocate resources to UE. According to Algorithm 1, we can get a conclusion as lemma 2 shows.

Lemma 2. *Through one-to-many matching between MEC server and AP in Algorithm 1, MEC server can obtain maximum utility.*

Proof. In Algorithm 1, MEC server constructs a preference list over all AP. A higher preference for AP represents that MEC server can gain greater outcomes if distributing CRBs to the AP. We set an indicator pointer of list to point at the most preferred AP initially. If there is rest resource that can be allocated after last round, the pointer moves to the second preferred AP in the list. Therefore, the pointer can only move on one direction. In other words, MEC server cannot achieve a higher utility through other allocating order. Thus, server achieves maximum utility.

After renting CRBs from MEC server, AP needs to reallocate them to UE if it cannot meet users' demand. Based on serving price, AP has difference preference over all UE, denoted as $N^k = \{\eta_1^k, \eta_2^k, \ldots, \eta_{M_k}^k\}$

$$\eta_i = r_i \quad (20)$$

According to relationship between AP and UE, we establish a one-to-many matching in Algorithm 2. The procedures are similar to that in Algorithm 1. Matching objects are the main difference between them. Since a real-time task is considered inseparably generally, it is worth noting that if number of CRBs that AP can supply is less than what UE needs, the UE will be abandoned, and pointer moves to next one as shown in steps 9–12. On the basis of Algorithm 2, the AP which can rent CRBs from MEC server is able to gain the best profits.

Simulation results and analysis

In this section, simulation results of the proposed scheme are presented with MATLAB. Generally, there are a MEC server and 10 APs, which are deployed randomly in a circle with diameter of 10 km. Each AP is connected to several UE located within a distance of 1 km. Amount of UE connected to one AP is 20 approximately. MEC server can supply 1000 CRBs to

Algorithm 1. One-to-Many Matching Algorithm between MEC server and AP

Construct preference list $L = \{L_1, L_2, \ldots, L_K\}$ for MEC server on all AP according to equation (19);
One pointer is set as an indicator pointing at first AP in preference list;
Construct a list $Q^a = \{Q_1, Q_2, \ldots Q_K\}$ to represent number of CRB that AP expects to rent from MEC server according to equation (18). List $Q^* = \{Q_1{}^*, Q_2{}^*, \ldots Q_K{}^*\}$ indicates real number of CRB that AP is allocated from MEC server, initialized as a zero profile;
Initialize total available number of CRB owned by MEC server $a_0 = N$;
Set a flag f_k to indicate whether AP a_k has been chosen by MEC server in last round; initially, $f_k = 1$; once a_k is chosen, $f_k = 0$;
MEC server determines which AP to choose according to preference list;
for a_k which is chosen by MEC server in last round **do**
    if $a_0 \geqslant Q_K$ **then**
        $a_0 = a_0 - Q_K, Q_K{}^* = Q_K$;
    else
        $Q_K{}^* = a_0, a_0 = 0$;
    end if
    Pointer moves to the most preferred AP in rest ones whose $f_k = 1$;
end for

Algorithm 2. One-to-Many Matching Algorithm between AP and UE

for a_k **do**
    Construct a preference list $N^k = \{\eta_1^k, \eta_2^k, \ldots, \eta_{M_k}^k\}$ for all UE according to equation (20);
    One pointer is set as an indicator pointing at first UE in preference list;
    Initialize total available number of CRB owned by a_k as $b_0 = Q_K{}^*$;
    Set a flag g_j to indicate whether s_j^k has been chosen by AP in last round; initially, $g_j = 1$; once s_j^k is chosen, $g_j = 0$;
    a_k determines which UE to choose;
    for s_j^k which is chosen by AP in last round **do**
        if $b_0 \geqslant q_j^k$ **then**
            $b_0 = b_0 - q_j^k$;
        else
            $q_j^k = b_0, b_0 = 0$;
        end if
        Pointer moves to most preferred UE in rest ones whose $g_j = 1$;
    end for
end for

users. According to Liu et al.,[24] the service rate of each computational block is set as 0.1 ms^{-1}, and workload arrival rate of each device is a random number averaged 0.5 ms^{-1}. And maximum tolerance of delay is 60 ms. Weight factors in utility function of UE are set as 50, 0.01, 0.001.

In simulation, experiments are conducted with different parameters changing, that is, total amounts of user devices, number of AP, workload arrival rate, service rate, and so on, to evaluate the proposed scheme in terms of utilities of different objects. Furthermore, we make a comparison between the proposed scheme and an allocation strategy based on auction game,[26] to evaluate performance and efficiency of the allocation mechanism.

As shown in Figure 2, utilities of MEC server and ratio of UE that can be served in two allocation strategies are compared with total number of equipment changing, ranging from 50 to 400. The simulation is carried out for 100 times. The figure describes that as amount of user devices rises, utility of MEC server increases gradually and coverages to an upper bound eventually. Schemes based on Stackelberg game outperform the situation under auction game all the time. The Stackelberg equilibrium ensures that the optimal CRB amounts and service price, chosen by UE, can maximize utility of MEC server. However, in auction game, UE is distributed fewer CRBs within maximum tolerance of delay, affecting profits that MEC server can receive. Utility gap will decrease as devices get more, as computing resource gets saturated. Nevertheless, the mechanism according to auction game is more stable with smaller variances.

On the contrary, ratio of UE that can be served under two curves both decline with equipment increasing. And compared with Stackelberg game, auction game can satisfy requirements of more UE. Due to CRBs amounts that UE requests are fewer, the server

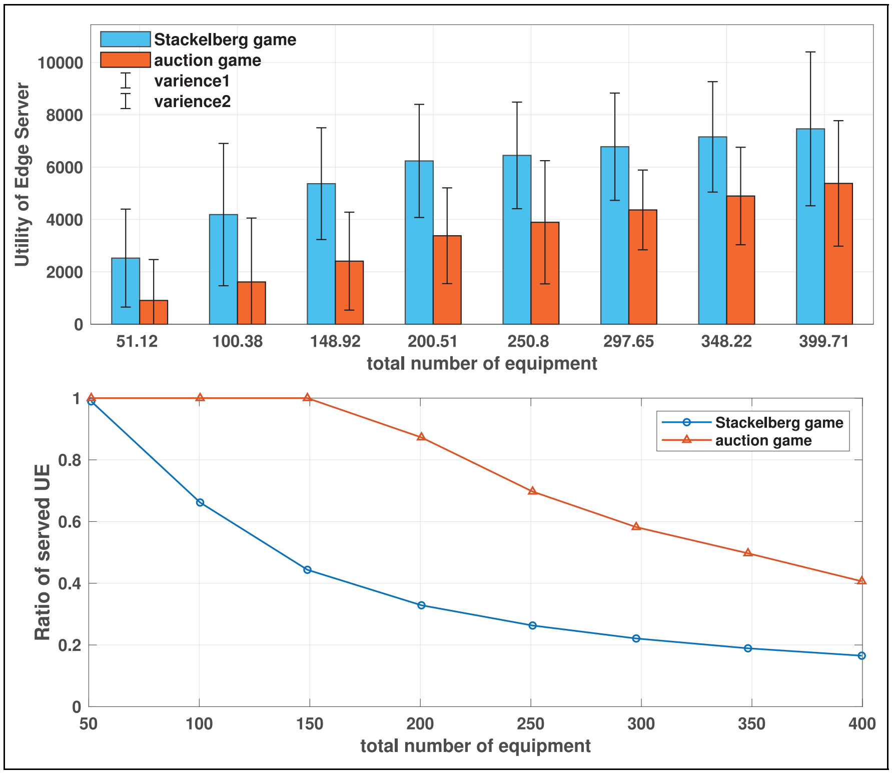

Figure 2. Comparison between schemes based on Stackelberg game or auction game.

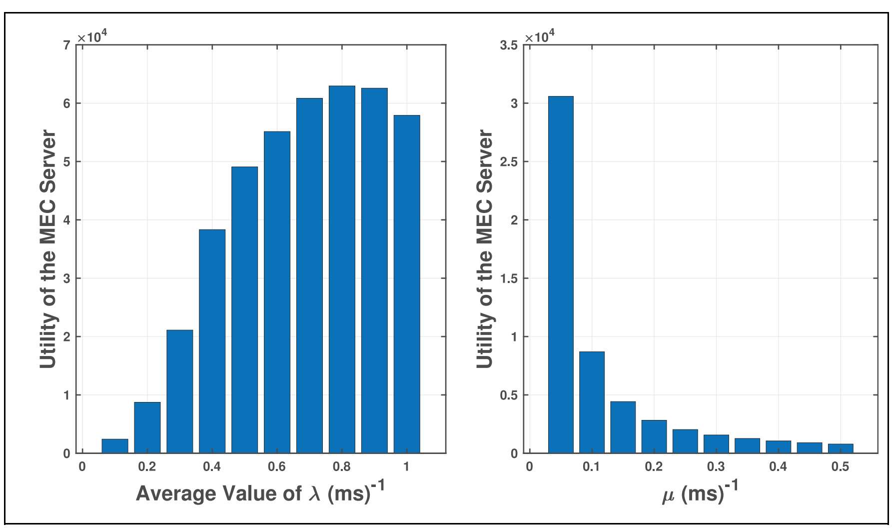

Figure 3. Utility of MEC server with different parameters.

can meet their demands with higher ratio based on auction game.

Figure 3 describes that utility of MEC server changes with different workload arrival rate, λ, and service rate of CRB, μ. The utility rises when λ increases. Higher workload rate causes that UE requires more CRBs to complete its tasks; hence, the server can allocate more CRBs and obtain higher profits until all CRBs are distributed. In addition, based on their distance, renting price that AP pays for MEC server remains unchanged. Therefore, incomes that MEC server receives are greater than expenditure.

With respect to μ, utility of MEC server decreases and coverages to zero ultimately as illustrated in Figure 3. A greater service rate means that UE can conduct computation tasks with fewer CRBs. And total

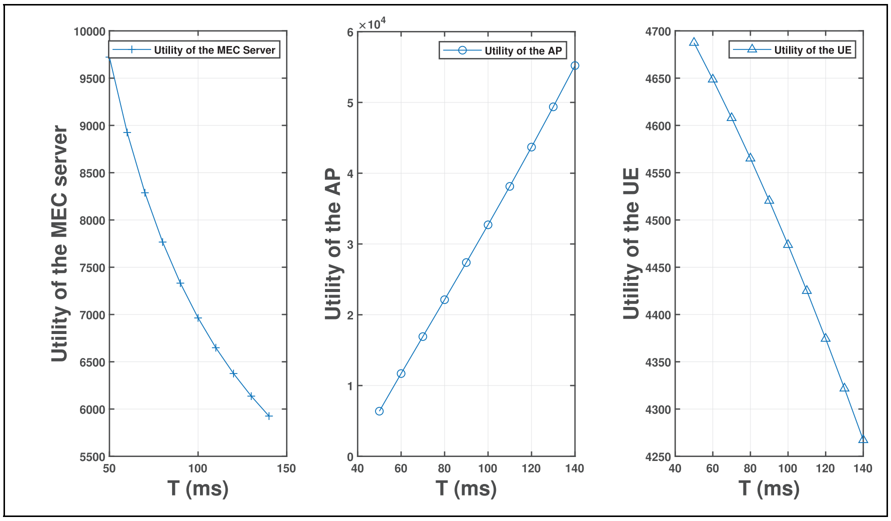

Figure 4. Utility with T changing.

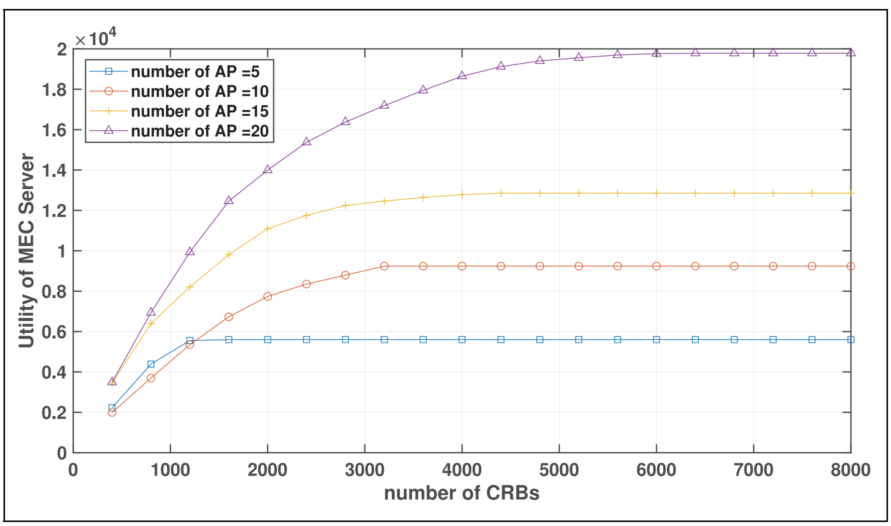

Figure 5. Utility of MEC server with different number of CRBs.

CRBs that all UE needs decline. Therefore, MEC server cannot gain the same revenues as before.

The maximum tolerance of delay, T, has various effects on utilities of MEC server, AP, and UE as indicated by Figure 4. First, when T increases, ranging from 40 to 140 ms, utility of server decreases and converges to a low bound. Since users can finish their tasks with fewer CRBs in a higher tolerance of delay, revenues that server can obtain from renting resources get lower. Second, utility of AP increases linearly and positively with delay increasing. According to equation (17), serving price is related to quadratic function T. With an unchanged renting price for MEC server and a higher pricing price for UE, AP gets more outcomes. Finally, utility of UE declines. Reasons are that even though optimal amount of CRBs decreases, serving price increases by a larger margin for UE as T gets higher. Therefore, UE receives fewer profits.

From Figures 5–7, utilities of MEC server, AP, and UE are shown as number of available CRBs increasing in four conditions, where different amounts of AP are deployed in network. Number of AP are 5, 10, 15, and 20, respectively. Figure 5 illustrates that utility of MEC server increases and reaches an upper bound finally when CRBs increase. Utility with more AP rises in faster speed. Figure 6 shows that as CRBs increase, utility rises and remains stable finally. In Figure 7, utility of UE is proportional to number of CRBs. To sum up, when CRBs increase, MEC server can supply computing services for more UE, and utility of all objects get higher. Once demands of UE in the framework become saturated, utilities increase to a certain point.

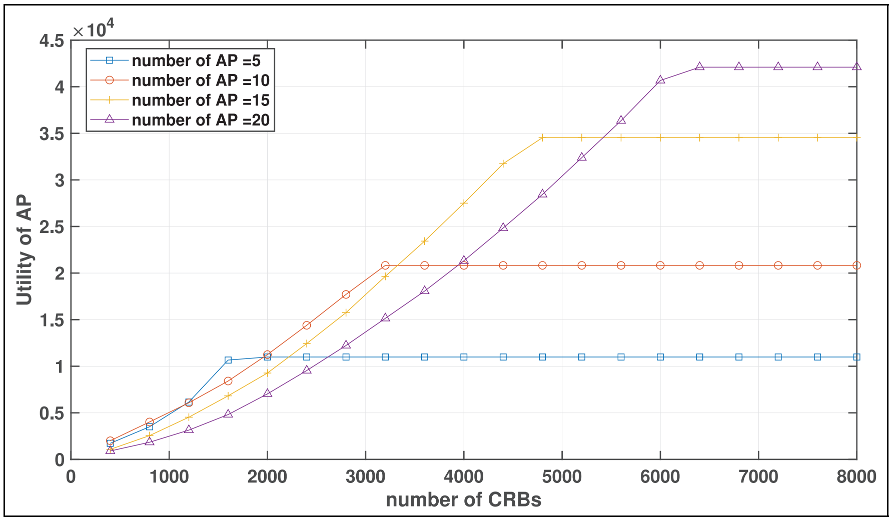

Figure 6. Utility of AP with different number of CRBs.

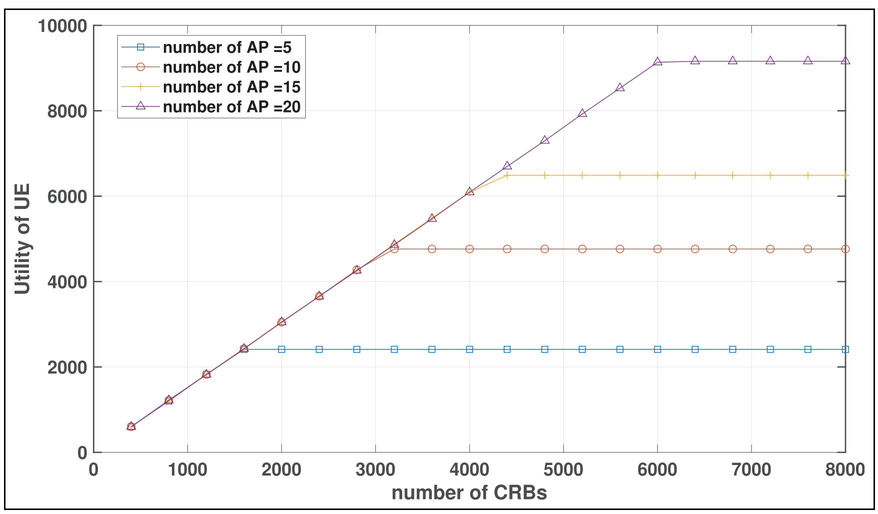

Figure 7. Utility of UE with different number of CRBs.

Furthermore, they all get higher utilities with more AP deployed. Analyzing these figures, we can conclude that when number of AP is 5, number of CRBs required by UE are about 1500. In other three conditions, total CRBs required are 3500, 4500 and 6000, respectively.

Conclusion and future work

In this article, we establish a hierarchical framework including MEC server, AP, and UE for Smart Home networks based on MEC. And an optimization problem is formulated, limited by computational resources, taking diverse requirements for delay and rate of users into consideration. Stackelberg game is proposed to solve resource purchasing problem and pricing problem for AP. Then, one-to-many matching is adopted to handle computational resource allocation problems for MEC server and AP. With maximum utility, the allocation strategy can improve allocating efficiency and guaranteeing real-time QoS for user terminals. Simulation results demonstrate the influence of different parameters such as number of CRBs and workload arrival rate. It is proved that compared with allocating mechanism based on auction game, the proposed framework can achieve a higher utility for server under Stackelberg equilibrium.

For the future work, we will take mobility and interactivity of smart terminals into account and apply the mechanism into smart city.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by National Natural Science Foundation of China (Grant No. 601702048) and Industrial Internet platform standard management service public support platform.

ORCID iD

Xing Hu https://orcid.org/0000-0002-5706-1483

References

1. Vallati C, Virdis A, Mingozzi E, et alet al. Mobile-edge computing come home connecting things in future smart homes using LTE device-to-device communications. *IEEE Consum Electron M* 2016; 5: 77–83.
2. Aun NFM, Soh PJ, Alhadi AA, et alet al. Revolutionizing wearables for 5G: 5G technologies: recent developments and future perspectives for wearable devices and antennas. *IEEE Microw M* 2017; 18: 108–124.
3. Networking CV. Cisco global cloud index: forecast and methodology. 2015–2020 White Paper, https://www.cisco.com/c/dam/m/en_us/service-provider/ciscoknowledge-network/files/622_11_15-16-Cisco_GCI_CKN_2015-2020_AMER_EMEAR_NOV2016.pdf
4. Baktir AC, Ozgovde A and Ersoy C. How can edge computing benefit from software-defined networking: a survey, use cases and future directions. *IEEE Commun Surv Tut* 2017; 19: 2359–2391.
5. Hui P, Ning H, Qiu T, et alet al. Fog computing based face identification and resolution scheme in Internet of things. *IEEE T Ind Inform* 2017; 13: 1910–1920.
6. Su Z, Hui Y, Xu Q, et alet al. An edge caching scheme to distribute content in vehicular networks. *IEEE T Veh Technol* 2018; 67: 5346–5356.
7. Tang B, Chen Z, Hefferman G, et alet al. Incorporating intelligence in fog computing for big data analysis in smart cities. *IEEE T Ind Inform* 2017; 13: 2140–2150.
8. Hua S. The present development situation and prospect of intelligent home. *China Comput Commun* 2017; 3: 153–154.
9. Rashidi P, Cook DJ, Holder LB, et alet al. Discovering activities to recognize and track in a smart environment. *IEEE T Knowl Data En* 2011; 23: 527–539.
10. Pocatilu P, Alecu F and Vetrici M. Measuring the efficiency of cloud computing for e-learning systems. *World Sci Eng Acad Soc* 2010; 9: 42–51.
11. Yu T, Wang X, Jin J, et alet al. Cloud-orchestrated physical topology discovery of large-scale IoT systems using UAVs. *IEEE T Ind Inform* 2018; 14: 2261–2270.
12. Perera C, Dumidu ST, Liu CH, et alet al. Energy-efficient location and activity-aware on-demand mobile distributed sensing platform for sensing as a service in IoT clouds. *IEEE T Comput Soc Syst* 2015; 2: 171–181.
13. Tao F, Cheng Y, Xu LD, et alet al. CCIoT-CMfg: cloud computing and Internet of things-based cloud manufacturing service system. *IEEE T Ind Inform* 2014; 10: 1435–1442.
14. Cai H, Xu B, Jiang L, et alet al. IoT-based big data storage systems in cloud computing: perspectives and challenges. *IEEE Internet Things* 2017; 4: 75–87.
15. Harmon E, Ozgur U, Hazar M, et alet al. The Internet of microgrids: a cloud-based framework for wide area networked microgrids. *IEEE T Ind Inform* 2018; 14: 1262–1274.
16. Wang W, Xu P and Yang LT. Secure data collection, storage and access in cloud-assisted IoT. *IEEE Cloud Comput* 2018; 5: 77–88.
17. Wang C, Yu FR, Liang C, et alet al. Joint computation offloading and interference management in wireless cellular networks with mobile edge computing. *IEEE T Veh Technol* 2017; 66: 7432–7445.
18. Munoz O, Pascualiserte A and Vidal J. Joint allocation of radio and computational resources in wireless application offloading. In: *2013 future network and mobile summit*, Lisboa, 3–5 July 2013, pp.1–10. New York: IEEE.
19. Zhang T, Chen W and Yang F. Data offloading in mobile edge computing: a coalitional game based pricing approach. *IEEE Access* 2018; 6: 2760–2767.
20. Lian F, Chakrabortty A and Duel HA. Game-theoretic multi-agent control and network cost allocation under communication constraints. *IEEE J Sel Area Comm* 2017; 35: 330–340.
21. Zhang L, Zhou X, Wang J, et alet al. Power control algorithm based on differential game for CR system. *J Electron Inform Technol* 2010; 32: 141–145.
22. Zhang H, Xiao Y, Bu S, et alet al. Computing resource allocation in three-tier IoT fog networks: a joint optimization approach combining Stackelberg game and matching. *IEEE Internet Things* 2017; 4: 1204–1215.
23. Attia II and Ashour H. Energy saving through smart home. *Online J Power Energy Eng* 2010; 2: 223–227.
24. Liu Z, Lin M, Wierman A, et alet al. Geographical load balancing with renewables. *Perf E R* 2011; 39: 62–66.
25. Myerson RB. *Game theory: analysis of conflict*. Beijing, China: Chinese People's Publishing House, 2015, pp.254–283.
26. Ausubel LM. An efficient dynamic auction for heterogeneous commodities. *Am Econ Rev* 2006; 96: 602–629.