Future Generation Computer Systems 108 (2020) 273–287

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

A Stackelberg game approach to multiple resources allocation and pricing in mobile edge computing

Yifan Chen [a,b], Zhiyong Li [a,b,*], Bo Yang [c], Ke Nai [a,b], Keqin Li [d]

[a] College of Computer Science and Electronic Engineering, Hunan University, Changsha, 410082, China
[b] Key Laboratory for Embedded and Network Computing of Hunan Province, Changsha, 410082, China
[c] College of Information and Management, Hunan University of Finance and Economics, Changsha, 410205, China
[d] Department of Computer Science, State University of New York, New Paltz, NY 12561, United States of America

ARTICLE INFO

Article history:
Received 30 April 2019
Received in revised form 18 February 2020
Accepted 18 February 2020
Available online 28 February 2020

Keywords:
Game theory
Mobile edge computing
Multiple resources allocation
Resource pricing

ABSTRACT

Mobile edge computing is a new paradigm that can enhance the computation capability of end devices and alleviate communication traffic loads during transmission. Mobile edge computing is highly useful for emerging resource-hungry mobile applications. However, a key challenge for mobile edge computing systems is multiple resources allocation between Mobile Edge Clouds (MECs) and End Users (EUs), especially for multiple heterogeneous MECs and EUs. To address this problem, we propose a Stackelberg game-based framework in which EUs and MECs act as followers and leaders, respectively. The proposed framework aims to compute a Stackelberg equilibrium solution in which each MEC achieves the maximum revenue while each EU obtains utility-maximized resources under budget constraints. We decompose the multiple resources allocation and pricing problem into a set of subproblems in which each subproblem only considers a single resource type. The Stackelberg game framework is constructed for each subproblem wherein each player (i.e., an EU) can selfishly maximize its utility by selecting an appropriate strategy in the strategy space. We prove the existence of the subgame Stackelberg equilibrium and develop algorithms to determine the Stackelberg equilibrium for each resource type, including an optimal demand computation algorithm, to determine the best resource demand strategy for an EU and an iterative algorithm to find an equilibrium price. The equilibrium solutions of all subgames constitute the equilibrium solution of the original problem. We also conduct simulation experiments of our game, such as numerical data for the Stackelberg equilibrium, numerical data for the convergence of the Stackelberg equilibrium, and numerical data as the system size increases. Finally, we demonstrate that an EU with idle resources can play the role of an MEC.

1. Introduction

Mobile edge computing is a novel computing paradigm that complements and effectively avoids several shortcomings of traditional clouds, such as reducing network traffic and enhancing user experience. Multiple types of resources, including computing, storage, control, and network resources, are often geographically close to End Users (EUs) in mobile edge computing. The processing capability of ordinary smart mobile devices can be greatly improved by taking advantage of mobile edge computing, which can even implement functions that are impossible with a laptop or a desktop. Various devices, such as smartphones, access points, and Base Stations (BSs), can act as mobile edge computing nodes or Mobile Edge Clouds (MECs) [1]. For example, a smartphone is an edge between an individual and the cloud, and a gateway in a smart home is the edge between home devices and the cloud. Moreover, edge computing is a key technology for meeting the stringent requirements of new systems and low-latency applications (e.g., embedded artificial intelligence, 5G networks, virtual/augmented reality, and tactile Internet).

Despite its great potential, mobile edge computing is still in its infancy and faces enormous new challenges such as programming models, network architecture design, Internet of things support, resource management and provisioning, resource allocation, security and privacy, and scalability of edge devices [1–3]. In this work, we focus on multiple resources allocation and pricing for a mobile edge computing system in which different EUs compete for resources in a resource pool composed of multiple MECs.

Compared with traditional clouds with virtually infinite capacities that are far from EUs, MECs have limited computational

* Corresponding author at: College of Computer Science and Electronic Engineering, Hunan University, Changsha, 410082, China.
E-mail addresses: cyf176@hnu.edu.cn (Y. Chen), zhiyong.li@hnu.edu.cn (Z. Li), bo_yang@hnu.edu.cn (B. Yang), naike_hnu@hnu.edu.cn (K. Nai).

https://doi.org/10.1016/j.future.2020.02.045
0167-739X/

power but are near EUs. Several MECs with different limited resources may exist in a geographic area, and each MEC may serve multiple EUs with endless sequences of computation tasks, various application characteristics, and diversified communication requirements and bandwidths. Therefore, multiple heterogeneous EUs compete for different resources from multiple heterogeneous MECs.

Resource allocation exhibits locality in mobile edge computing. An MEC can only serve its neighboring EUs with limited computing resources, whereas EUs with different requirements can only offload computing tasks to their proximate MECs. In this manner, when an EU has high-priority emergency tasks, such as e-health, mobile vehicle connectivity, and industrial automation, the nearby MEC cannot meet the needs of the EU because it can only allocate limited computing resources to the EU. This limitation becomes more prominent when the number of EUs who opt to offload their tasks grows. Moreover, EUs and MECs genetically belong to different authorities, are profit-oriented, and focus only on their own utilities. Therefore, a feasible and efficient incentive mechanism is required to charge EUs and reward MECs to stimulate service provisioning by MECs and improve resource utilization. When the mobile edge computing system adopts price incentive strategies, MECs can also gain certain payoffs while providing resources to EUs. EUs can obtain a superior user experience while paying for the resources.

EUs inherently expect to obtain as many resources as possible within their limited budgets, and MECs with a limited amount of resources aim to maximize their revenues by attracting EUs to purchase their resources. However, realizing these requirements is difficult in distributed systems, such as mobile edge computing systems, where no centralized authority exists, and MECs and EUs can selfishly make their own decisions.

Therefore, a primary concern is the efficient allocation of limited MEC resources to competing EUs with diverse demands and preferences. To address this challenge, we propose a Stackelberg game-based framework to harmonize the interests of EUs and MECs such that every EU is satisfied with its obtained resources, while MECs maintain considerable revenue. The main contributions of this work are summarized as follows:

(1) We establish a Stackelberg game model for the multiple resources allocation and the pricing problem between multiple EUs and MECs. In our model, each EU with a limited budget who acts as a follower can decide the demand matrix, which represents its requirements for each resource owned by each MEC to maximize its utility. Each MEC with constrained resources that acts as a leader can determine the unit price of various resources to maximize its revenue.
(2) We decompose the multiple resources allocation and pricing problem into a set of subproblems in which each subproblem only considers a single resource type. The Stackelberg game framework is constructed for each subproblem in which leaders (i.e., MECs) can determine the unit price of that resource, and each follower (i.e., EU) can selfishly maximize its utility by selecting an appropriate strategy in the strategy space. Moreover, we prove the existence of the Stackelberg equilibrium of the subproblem game.
(3) We develop algorithms to determine the Stackelberg equilibrium for each type of resource, including an Optimal Demand Computation Algorithm (ODCA) to find the best resource demand strategy for an EU and an iterative algorithm to find an equilibrium price. The equilibrium solutions of all subgames constitute the equilibrium solution of the original problem.
(4) We conduct the simulation experiments of our game, including the numerical data for the Stackelberg equilibrium and the numerical data for the convergence of the Stackelberg equilibrium. Furthermore, we show that an EU with idle resources can play the role of an MEC.

The remainder of this work is organized as follows. We describe the related work in Section 2 and discuss the system model in Section 3. We formulate a Stackelberg game for multiple EUs that compete for multiple resources from multiple heterogeneous MECs, show the existence of the Stackelberg equilibrium of the game, and develop algorithms to find the Stackelberg equilibrium in Section 4. We investigate the simulation results in Section 5. Finally, we conclude the paper in Section 6.

2. Related works

Mobile edge computing and related technical works have obtained much attention in recent years. The latest comprehensive surveys provide overviews of most of these published works [4–6].

One major line of research has recently focused on computing offloading from a user equipment to an MEC [7–10]. The common characteristic among the literature is that EUs offload their computational tasks to MECs to reduce their power consumption or computational task execution delays. Typically, these MECs with limited resources are located near the EUs or at cellular BSs. However, multiple EUs offloading their computation tasks to MECs simultaneously may negatively affect the EUs' experience for computing offloading. The reason is that an EU may suffer from the following problems: increased transmission delay due to severe communication interference, high task processing delay caused by more waiting time in a queuing system, and more power consumption due to increased transmission delay. Therefore, the allocation and scheduling of resources should be considered in the mobile edge computing system.

Much research concerning cloud resource allocation and pricing has been conducted [11]. A widely used method is a game-theoretical approach to tackle resource provisioning and pricing problems in cloud computing [12–16]. Liu et al. focused on request migration strategies among multiple servers for load balancing from a game-theoretic perspective and formulated it into a non-cooperative game among the multiple servers, where it is under a distributed, non-cooperative, and competitive environment [15]. Auction theory is another state-of-the-art approach focused on the pricing problem [17–20]. In [17], the authors proposed a combinatorial auction-based mechanism to address Virtual Machine (VM) allocation and pricing in the presence of multiple types of VMs in a single-provider scenario.

By contrast, resource allocation and pricing in mobile edge computing is still in its infancy [21–26]. Mao et al. investigated an online joint radio and computational resource management algorithm for multi-user mobile edge computing systems to minimize the long-term average weighted sum power consumption of the mobile devices and the mobile edge computing server, subject to a task buffer stability constraint [21]. You et al. studied resource allocation for a multi-user mobile edge computation offloading system based on time-division multiple access and orthogonal frequency-division multiple access, in which each user has one task, by minimizing the weighted sum of mobile energy consumption under the constraint on computation latency, with the assumption of negligible cloud computing and result downloading time [22]. Guo et al. considered energy-efficient resource allocation schemes for a multi-user mobile edge computing system with one BS of computing capability and multiple users with inelastic computation tasks of non-negligible task execution durations by minimizing the weighted sum energy consumption problem to optimally allocate the task operation sequence as well as the uploading and downloading time durations [23]. Yin et al. proposed a container-based task-scheduling model and task-scheduling algorithms with a task-delay constraint. Furthermore, in accordance with the characteristics of the container, a resource-reallocation mechanism was proposed to reduce the execution delay of tasks [24]. These studies typically discuss resource management issues to minimize power consumption or reduce task completion time. However, these works consider neither the pricing of resources nor the issue of simultaneously optimizing the goals of EUs and MECs.

A group of literature utilizes auction theory to investigate resource pricing in mobile edge computing. Jin et al. considered resource sharing and pricing for cloudlets in mobile cloud computing in [27] and proposed two incentive mechanisms, including a truthful incentive mechanism, and a more efficient design of auction to coordinate the resource auction between mobile devices as service users (buyers) and cloudlets as service providers (sellers). In [28], Jin et al. proposed an incentive-compatible auction mechanism for the resource trading between the mobile devices as service users (buyers) and cloudlets as service providers (sellers). Sun et al. investigated the joint problem of network economics and resource allocation in mobile edge computing where industrial Internet of Things (IIoT) Mobile Devices (MDs) requested offloading with claimed bids, and edge servers provided their limited computing service with ask prices. They proposed two double auction schemes with dynamic pricing in mobile edge computing, namely, a breakeven-based double auction and a more efficient dynamic pricing-based double auction, to determine the matched pairs between IIoT MDs and edge servers as well as the pricing mechanisms for high system efficiency under the locality constraints [29].

Matching theory is a mathematical framework that provides polynomial-time solutions for combinatorial assignment problems [30]. Matching theory is suitable to model the interactions among numerous agents with conflicting interests [31]. For example, in [32], a distributed spectrum trading mechanism based on matching theory with evolving preferences is introduced to provide more spectrum accessing opportunities for secondary users through spectrum reuse. In [33], the authors studied caching in dual-mode Small Base Stations (SBSs) that integrate μ-wave and mm-wave frequencies, where a dynamic matching game-theoretic approach was applied to maximize the handovers to SBSs in the mobility management scenarios. However, the research in this paper explores not only the combinatorial assignment problem but also multiple resources pricing. As such, matching theory is unsuitable for the proposed model.

A Stackelberg game is a two-period game with a concept of leader and follower [34]. The leader and follower try to maximize their profits. Hence, the game provides dual benefits to both players. A Stackelberg game has been extensively adopted for solving resource management problems in a network system with a distributed fashion. In [35], Wang et al. formulated a Stackelberg game for the power allocation of data centers in the cloud. In the game, the power grid controller acts as the leader and sets the prices of the provided energy based on the current amount of renewable energy and costs. The cloud controller, i.e., the follower, observes the prices, determines the optimal amount of energy to purchase, and performs resource allocation for its data centers. The near-optimal strategies of both players in the game can be achieved using backward induction. A Stackelberg game-theoretic model is widely used to describe hierarchical decision-making problems in a network system [36–40]. In [36], Zhang et al. considered a specific fog computing network consisting of a set of Data Service Operators (DSOs), each of which controls a set of Fog Nodes (FNs) to provide the required data service to a set of Data Service Subscribers (DSSs). They formulated a Stackelberg game to analyze the pricing problem for the DSOs as well as the resource allocation problem for the DSSs. They proposed a many-to-many matching between the DSOs and the FNs to deal with the DSO-FN pairing problem. Reference [37] formulated the single-cloud multi-service resource provisioning and pricing problem as a Stackelberg game to minimize the services' costs while maximizing the provider's revenue. Yang et al. employed game-theoretic approaches to model the problem of minimizing energy consumption as a Stackelberg game [38]. Several related studies on resource allocation and pricing in mobile edge computing have recently used a Stackelberg game method [41–43]. For example, in [41], Guo et al. proposed a hierarchical architecture in Smart Home with mobile edge computing and adopted a Stackelberg game to solve resource purchasing and pricing problem for access point and user equipment. Therefore, a Stackelberg game is an appropriate model for solving the problem proposed in this paper.

However, as shown in Table 1, the previously mentioned works are either based on centralized methods or deal with single resource allocation problems in a distributed environment. Different from their considerations, to address the problem of multiple resources allocation and pricing in a distributed environment, we propose a multi-leader multi-follower Stackelberg game to harmonize the interests of MECs and EUs. Our paper has the following novel and unique features.

(1) We consider multiple heterogeneous EUs competing for multiple resources from multiple heterogeneous MECs, in which each EU is limited by the budget for each resource, and each MEC has a certain amount constraint for each type of resource. The interaction between EUs and MECs is carried out by the price incentive mechanism.
(2) Each EU has a budget to procure resources, which includes budgets for each required resource. Each MEC contains several types of resources, and the number of each resource is limited.
(3) We employ a Stackelberg game approach to address the optimal resource demands for each EU and the unit price of each resource for MECs.

3. System model

We introduce the system model of mobile edge computing in this section. Fig. 1 depicts a network architecture with three layers, namely, the MEC layer, the aggregation layer, and the EU layer. First, various devices, such as smartphones, tablets, PCs, lab servers, and underutilized small and medium data centers in enterprises/schools/hospitals and central telecom offices, can act as MECs in the MEC layer. In general, equipment with certain computing capabilities and is temporarily idle can act as MEC. Second, the aggregation layer (e.g., a base station, a switch/router, or an access point) is the communication bridge between EUs and MECs. Third, the EU layer consists of diverse equipment, such as sensors, smartphones, or PCs, that have different computation requirements.

Each EU has several types of computing tasks and may require several types of resources. This paper assumes that each EU has a definitive budget for a specific resource. In practice, the budgets for these resources can be transferred to one another. However, we do not consider this situation to simplify the problem. Correspondingly, each MEC also contains at least one type of resource. The amount of each type of resource is limited. Moreover, MECs can be paid for providing their resources to the EUs. In general, EUs and MECs exchange price and demand information through the aggregation layer, as shown in Fig. 1. All MECs publish the unit price of their own resources, and EUs transmit their demand

Table 1
Comparisons between ours and the state-of-the-art works.

Ref.	Environment	Number of resource type	Main technique(s)
[21]	Central	Multi-resources	Lyapunov optimization
[26]	Central	Two	Convex optimization and heuristic methods
[28]	Distributed	Multi-resources	Auction
[29]	Distributed	Multi-resources	Double auction
[41]	Distributed	Single resource	Stackelberg game and matching strategy
[42]	Distributed	Single resource	Multi-leader multi-follower Stackelberg game
[43]	Distributed	Single resource	Single-leader multi-follower Stackelberg game
Ours	Distributed	Multi-resources	Multi-leader multi-follower Stackelberg game

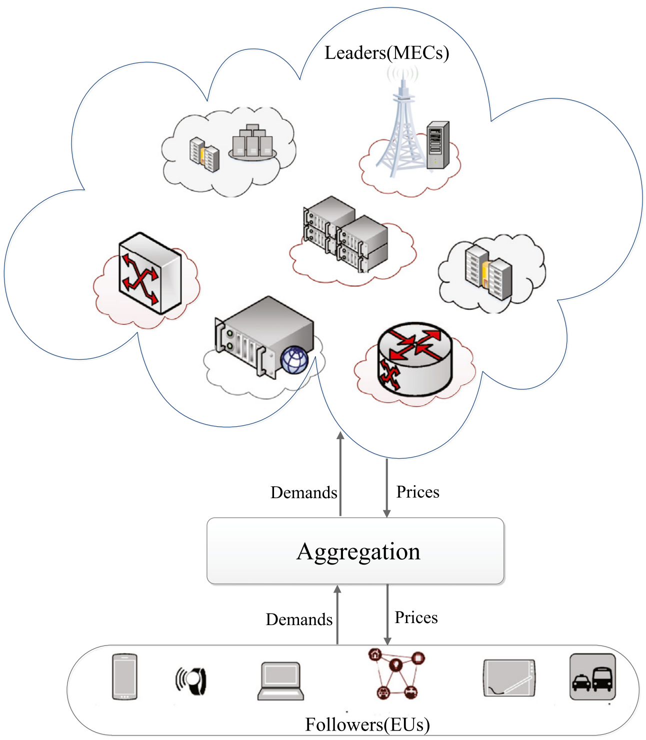


Fig. 1. Mobile edge computing system with multiple EUs and MECs. EUs and MECs exchange price and demand information through the aggregation layer.

to MECs based on the unit price of each resource and their budget for each resource.

The mobile edge computing system in this work consists of multiple EUs and multiple resource-constrained heterogeneous MECs. Each MEC provides several types of resources. For example, an MEC can be selected randomly from a set of M4 and M5 Amazon EC2 instances [44]. An M4 Amazon EC2 machine provides multiple configurations of resource types, such as m4.large, m4.xlarge, and m4.2xlarge. Each EU has an available budget to buy resources. These resources are utilized by EUs to perform computing tasks in the mobile edge computing system. We focus on the interactions between MECs and EUs in this paper and consider multiple heterogeneous EUs that compete for resources from multiple heterogeneous MECs.

Several mechanisms stimulate the collaboration between the MECs and the EUs. An incentive mechanism to charge EUs and to reward MECs is required. That is the unit price of each resource. On the one hand, if the price is high, then it can motivate the MECs to provide resources to the EUs. On the other hand, if the price is low, then it can promote EUs to purchase more resources from the MECs. In this paper, we focus on computing an equilibrium solution that assigns a unit price to each resource and allocates an optimal resource bundle to each EU to maximize the utilities of the MECs and EUs.

Table 2
Summary of key notations.

Notation	Description
(a) Sets	
$\mathscr{M}$	Set of MECs
$\mathscr{N}$	Set of EUs
$\mathscr{R}$	Set of resources
(b) Parameters	
M	Total number of MECs
N	Total number of EUs
R	Number of categories for different resources
$B_{i,r}$	Budget of EU i for resource r
B_i	Total budget of EU i for multiple resources
$Q_{j,r}$	Amount of resource r in MEC j
(c) Decision variables	
$x_{i,j,r}$	Number of resource of type r obtained by EU i from MEC j
$\boldsymbol{x_{i,j}}$	Demand vector of EU i for resources from MEC j
$\boldsymbol{X_i}$	Demand matrix of EU i for resources from all MECs
$p_{j,r}$	Unit price of resource r in MEC j
$\boldsymbol{p_j}$	Unit price vector of resources in MEC j
$\boldsymbol{P}$	Unit price matrix of all MECs

4. Problem formulation

As described in Fig. 1, MECs set the unit price for each type of resource and announce them to EUs. Afterward, EUs determine their demands for each resource based on the unit prices from MECs. MECs make choices before EUs, and this model is referred to as the Stackelberg model in economics [45].

The Stackelberg model is frequently used to model players in which a leader exists. For example, Apple is typically considered a dominant company in the mobile phone industry, where small companies commonly wait for Apple's release of new products. After the release, the small companies adjust their product decisions accordingly. In this example, we model the mobile phone industry with Apple as the Stackelberg leader and the other companies in the industry as the Stackelberg followers.

In our proposed model, the MECs are leaders who can set their prices, and the EUs are followers who respond by selecting their demands. This game consists of multiple leaders with multiple resources and followers. What price should the leader MECs set to maximize their utility? The answer depends on how the leader MECs expect the follower EUs to react to their prices. Presumably, leader MECs will expect that the follower EUs will attempt to maximize utilities as well. Therefore, leaders MECs must consider their followers EUs' utility maximization problem to make sensible decisions about their resource prices.

We must establish mathematical models to analytically study the details of the theoretical Stackelberg model. For ease of reference, Table 2 lists the key notations and descriptions adopted in this paper.

4.1. EU problems

Our research focuses on a mobile edge computing system, which includes multiple EUs and MECs with different types of resources (Fig. 1). We denote $\mathscr{N} = \{1, 2, \ldots, N\}$ and $\mathscr{M} = \{1, 2, \ldots, M\}$ as the set of EUs and MECs, respectively. The notation $\mathscr{R} = \{1, 2, \ldots, R\}$ represents the set of the different types of resources. Assume that $|\mathscr{N}| = N$, $|\mathscr{M}| = M$, and $|\mathscr{R}| = R$. B_i denotes the budget of EU i, which includes the budget for each resource r $B_{i,r}$, where $B_i = B_{i,1} + B_{i,2} + \cdots + B_{i,R}$. $B_{i,r}$ is 0 if EU i has no demand for resource r; otherwise, $B_{i,r} > 0$. Let $x_{i,j,r}$ be the amount of resource type r obtained by EU i from MEC j. The utility functions of the EUs are also provided. EU i aims to maximize the following function.

$$U_i = \sum_{r \in \mathscr{R}} B_{i,r} \sum_{j \in \mathscr{M}} \log(\alpha_i + x_{i,j,r}), \forall i \in \mathscr{N}, \tag{1}$$

where α_i is a constant. The log function has been widely used in network optimization problems, and equilibrium allocations for the Fisher market [46,47]. The function in Eq. (1) is selected as a utility function due to its close relation with the utility function $\omega_{i,r} \sum_{j \in \mathscr{M}} \log(x_{i,j,r})$. Furthermore, this function leads to a proportionally fair resource allocation [39,48], where $\omega_{i,r}$ is a constant. However, if $\omega_{i,r} \sum_{j \in \mathscr{M}} \log(x_{i,j,r})$ is used as the utility function, then an EU will obtain a negative utility when $x_{i,j,r} = 0$, which is inconsistent with the definition that the utility will be 0 when $x_{i,j,r} = 0$. After replacing $x_{i,j,r}$ with $\alpha_i + x_{i,j,r}$, the utility becomes finite when $x_{i,j,r} = 0$. The general value of α_i is 1. In addition, we use budget $B_{i,r}$ instead of $\omega_{i,r}$.

The follower EU's utility depends on resource matrix $\boldsymbol{X_i}$, which varies with the unit price matrix P of the leader MECs, where $\boldsymbol{X_i} = (\boldsymbol{x_{i,1}}, \boldsymbol{x_{i,2}}, \ldots, \boldsymbol{x_{i,M}})^T \in R^{M\times R}$, and the resource matrix, in which the element at the jth row and rth column is $x_{i,j,r}$. For a given price matrix $\boldsymbol{P}$ from the MECs, where $\boldsymbol{P} = (\boldsymbol{p_1}, \boldsymbol{p_2}, \ldots, \boldsymbol{p_M})$ and $\boldsymbol{p_j} = (p_{j,1}, p_{j,2}, \ldots, p_{j,R})$, EU i aims to select an optimal resource demand matrix $\boldsymbol{X_i}$ by solving the EU optimization problem

$$\max_{\boldsymbol{X_i}} \quad U_i \tag{2}$$

$$s.t. \quad \sum_{r \in \mathscr{R}} \sum_{j \in \mathscr{M}} p_{j,r} x_{i,j,r} \le B_i \tag{3}$$

$$x_{i,j,r} \ge 0, \forall r \in \mathscr{R}, \forall j \in \mathscr{M}. \tag{4}$$

The resource demand choices of each EU will depend on the prices determined by the MECs. Thus, this relationship can be written as

$$\boldsymbol{X_i} = f(\boldsymbol{P}), \forall i \in \mathscr{N}. \tag{5}$$

Function $f(\boldsymbol{P})$ reveals the choices of an EU as a function of the MECs' determination. This relationship is a reaction function because it reveals how EUs will respond to MECs' determination.

We derive the reaction function in the following discussion. We can decompose the original EU optimization problem into several single resource subproblems because all resources are independent of and not in conflict with one another. The EU optimization problem can be transformed into R single resource subproblems according to the number of resource type R. The subproblem r is expressed as

$$\max_{x_i^r := \{x_{i,j,r}, j \in \mathscr{M}\}} \quad U_{i,r} = B_{i,r} \sum_{j \in \mathscr{M}} \log(\alpha_i + x_{i,j,r}) \tag{6}$$

$$s.t. \quad \sum_{j \in \mathscr{M}} p_{j,r} x_{i,j,r} \le B_{i,r} \tag{7}$$

$$x_{i,j,r} \ge 0, \forall j \in \mathscr{M}. \tag{8}$$

Eq. (5) is another constraint, where $r = 1, \ldots, R$.

We present the following statements based on the above transformation.

(1) This transformation is equivalent because the x_i^r of all subproblems constitute the $\boldsymbol{X_i}$ of the original EU optimization problem.

(2) The solution of the original EU optimization problem is an $M \times R$ matrix. However, the solution is only a vector with M elements in a subproblem. Solving the subproblems is simpler than solving the original problem.

First, let us derive a reaction function in the simple case of N EUs and three MECs. Then, we apply the derived function to the general situation of N EUs and M MECs. In this case, subproblem r is defined as

$$\max_{x_i^r := \{x_{i,1,r}, x_{i,2,r}, x_{i,3,r}\}} \quad U_{i,r} = B_{i,r} \sum_{j=1}^{3} \log(\alpha_i + x_{i,j,r}) \tag{9}$$

$$s.t. \quad \sum_{j=1}^{3} p_{j,r} x_{i,j,r} \le B_{i,r} \tag{10}$$

$$x_{i,1,r}, x_{i,2,r}, x_{i,3,r} \ge 0. \tag{11}$$

The problem defined by Eqs. (9)–(11) always has a feasible interior solution by simply setting $x_{i,1,r}$, $x_{i,2,r}$, and $x_{i,3,r}$ as positive and sufficiently small, such that all constraints in Eqs. (10) and (11) are satisfied with the strict inequalities. Hence, the Slaters condition holds, and the Karush–Kuhn–Tucker (KKT) conditions are necessary and sufficient for optimality [49]. Inspired by studies [3] and [40], we also adopt the KKT conditions to solve the problem.

We define $\lambda_{i,r}$, $\mu_{i,1,r}$, $\mu_{i,2,r}$, and $\mu_{i,3,r}$ as the dual variables associated with Eqs. (10) and (11). The Lagragian function is expressed as

$$\begin{aligned} L(x_i^r, \lambda_{i,r}, \mu_{i,1,r}, \mu_{i,2,r}, \mu_{i,3,r}) &= B_{i,r} \sum_{j=1}^{3} \log(\alpha_i + x_{i,j,r}) \\ &+ \lambda_{i,r}(B_{i,r} - \sum_{j=1}^{3} p_{j,r} x_{i,j,r}) + \mu_{i,1,r} x_{i,1,r} + \mu_{i,2,r} x_{i,2,r} \\ &+ \mu_{i,3,r} x_{i,3,r}. \end{aligned} \tag{12}$$

The KKT conditions of the problem in Eqs. (9)–(11) are presented as follows.

$$\frac{\partial L}{\partial x_{i,1,r}} = \frac{B_{i,r}}{\alpha_i + x_{i,1,r}} - \lambda_{i,r} p_{1,r} + \mu_{i,1,r} = 0 \tag{13}$$

$$\frac{\partial L}{\partial x_{i,2,r}} = \frac{B_{i,r}}{\alpha_i + x_{i,2,r}} - \lambda_{i,r} p_{2,r} + \mu_{i,2,r} = 0 \tag{14}$$

$$\frac{\partial L}{\partial x_{i,3,r}} = \frac{B_{i,r}}{\alpha_i + x_{i,3,r}} - \lambda_{i,r} p_{3,r} + \mu_{i,3,r} = 0 \tag{15}$$

$$\lambda_{i,r}(B_{i,r} - \sum_{j=1}^{3} p_{j,r} x_{i,j,r}) = 0 \tag{16}$$

$$\mu_{i,1,r} x_{i,1,r} = 0 \tag{17}$$

$$\mu_{i,2,r} x_{i,2,r} = 0 \tag{18}$$

$$\mu_{i,3,r} x_{i,3,r} = 0 \tag{19}$$

$$\lambda_{i,r},\ \mu_{i,1,r},\ \mu_{i,2,r},\ \mu_{i,3,r},\ x_{i,1,r},\ x_{i,2,r},\ x_{i,3,r} \ge 0. \tag{20}$$

The optimal solutions of Eqs. (9)–(11) can take one of the following forms:

(1) Case 1: $x_{i,1,r} = x_{i,2,r} = x_{i,3,r} = 0$: if $B_{i,r} > 0$, then we can derive $\lambda_{i,r} = 0$ using Eq. (16). $\mu_{i,2,r}$ and $\mu_{i,3,r}$ must be

negative according to Eqs. (14) and (15), respectively, which do not satisfy Eq. (20). Thus, the solution in this situation is not feasible and cannot occur. However, when $B_{i,r} = 0$, verifying that $x_{i,1,r} = x_{i,2,r} = x_{i,3,r} = 0$ is easy.

(2) Case 2: $x_{i,1,r}, x_{i,2,r}, x_{i,3,r} > 0$: from Eqs. (17), (18), and (19), we can derive $\mu_{i,1,r} = \mu_{i,2,r} = \mu_{i,3,r} = 0$. Then, Eqs. (13)–(16) can be rewritten as follows.

$$\frac{B_{i,r}}{\alpha_i + x_{i,1,r}} - \lambda_{i,r} p_{1,r} = 0 \tag{21a}$$

$$\frac{B_{i,r}}{\alpha_i + x_{i,2,r}} - \lambda_{i,r} p_{2,r} = 0 \tag{21b}$$

$$\frac{B_{i,r}}{\alpha_i + x_{i,3,r}} - \lambda_{i,r} p_{3,r} = 0 \tag{21c}$$

$$\lambda_{i,r}(B_{i,r} - \sum_{j=1}^{3} p_{j,r} x_{i,j,r}) = 0. \tag{21d}$$

The above expressions signify that $\lambda_{i,r} > 0$. From Eqs. (21a)–(21d), we can derive

$$x_{i,j,r} = \frac{B_{i,r} + \alpha_i \sum_{j=1}^{3} p_{j,r}}{3p_{j,r}} - \alpha_i, \; j = 1, 2, 3. \tag{22}$$

(3) Case 3: $x_{i,1,r} = 0, x_{i,2,r}, x_{i,3,r} > 0$: Eqs. (18) and (19) imply that $\mu_{i,2,r} = \mu_{i,3,r} = 0$. On the basis of this knowledge, Eqs. (13)–(16) can be rewritten as follows.

$$\frac{B_{i,r}}{\alpha_i} - \lambda_{i,r} p_{1,r} + \mu_{i,1,r} = 0 \tag{23a}$$

$$\frac{B_{i,r}}{\alpha_i + x_{i,2,r}} - \lambda_{i,r} p_{2,r} = 0 \tag{23b}$$

$$\frac{B_{i,r}}{\alpha_i + x_{i,3,r}} - \lambda_{i,r} p_{3,r} = 0 \tag{23c}$$

$$\lambda_{i,r}(B_{i,r} - p_{2,r} x_{i,2,r} - p_{3,r} x_{i,3,r}) = 0. \tag{23d}$$

Suppose $\lambda_{i,r} = 0$. Then, $\frac{B_{i,r}}{\alpha_i + x_{i,3,r}} = 0$, which will never occur. Thus, $\lambda_{i,r} > 0$. From Eq. (21), we can derive the following expressions.

$$x_{i,2,r} = \frac{B_{i,r} + \alpha_i(p_{2,r} + p_{3,r})}{2p_{2,r}} - \alpha_i \tag{24a}$$

$$x_{i,3,r} = \frac{B_{i,r} + \alpha_i(p_{2,r} + p_{3,r})}{2p_{3,r}} - \alpha_i. \tag{24b}$$

(4) Case 4: $x_{i,2,r} = 0, x_{i,1,r}, x_{i,3,r} > 0$: This case is similar to Case 3.

(5) Case 5: $x_{i,3,r} = 0, x_{i,1,r}, x_{i,2,r} > 0$: This case is also similar to Case 3.

(6) Case 6: $x_{i,1,r} = x_{i,2,r} = 0, x_{i,3,r} > 0$: Proving that $\lambda_{i,r} > 0$ is easy. From Eq. (16), we can derive

$$x_{i,3,r} = \frac{B_{i,r}}{p_{3,r}}. \tag{25}$$

(7) Case 7: $x_{i,2,r} = x_{i,3,r} = 0, x_{i,1,r} > 0$: $x_{i,1,r} = \frac{B_{i,r}}{p_{1,r}}$: This case is similar to Case (6).

(8) Case 8: $x_{i,1,r} = x_{i,3,r} = 0, x_{i,2,r} > 0$: $x_{i,2,r} = \frac{B_{i,r}}{p_{2,r}}$: This case is also similar to Case (6).

In the cited case with N EUs and three MECs, the optimal demand strategy will be one of the eight cases that depends on the prices of the MECs and budget $B_{i,r}$. However, the strategy will become increasingly complex as the values of N and M increases. In the following discussion, we present an ODCA to rapidly compute the optimal demands.

Basing on Eqs. (22), (24), and (25), the demands for the general case of N EUs and M MECs for a given set of $\{p_{j,r}, j \in \mathscr{M}\}$, can be formulated as

$$x_{i,j,r} = \begin{cases} \frac{B_{i,r} + \alpha_i \sum_{t \in T} p_{t,r}}{|T| p_{j,r}} - \alpha_i \\ 0 \end{cases}, \tag{26}$$

where T is a set that must contain j, and $|T|$ represents the number of elements in T. $x_{i,j,r}$ is equal to either $(B_{i,r} + \alpha_i \sum_{t \in T} p_{t,r})/(|T| p_{j,r}) - \alpha_i$ or 0, and T is crucial to $x_{i,j,r}$. Initially, T is set by $\mathscr{M}$. Suppose $x_{i,j,r}$ is a real number. In this situation, we can obtain $x_{i,j,r} = (B_{i,r} + \alpha_i \sum_{j=1}^{M} p_{j,r})/(Mp_{j,r}) - \alpha_i, j \in \mathscr{M}$, which is similar to Eq. (22). However, several of $x_{i,j,r}$ may be negative because the MECs' $p_{j,r}$ are excessively high and the budget $B_{i,r}$ is relatively small. This situation contradicts the fact that no EU will have a negative demand. $x_{i,j,r}$ will then be reset to 0 when the computed $x_{i,j,r}$ from $x_{i,j,r} = (B_{i,r} + \alpha_i \sum_{j=1}^{M} p_{j,r})/(Mp_{j,r}) - \alpha_i$ is negative. In addition, we exclude j from T. Finally, we repeat the above mentioned process using the new T until all $x_{i,j,r}$ satisfy the condition $x_{i,j,r} \geq 0$.

In the above analysis, we suppose that all MECs have R resources and $p_{j,r} > 0, \forall j \in \mathscr{M}, r \in \mathscr{R}$. However, each MEC j contains at most R types of resources. Some MECs do not contain certain types of resources. MEC j will set the price to 0 when no resource r exists, that is, $p_{j,r} = 0$. In this case, $x_{i,j,r} = 0, \forall i \in \mathscr{N}$ must occur. Furthermore, this situation will have no influence on the demands of EUs for MECs, who own the resources.

Basing on the analysis above, we propose an ODCA for determining EU i's best demand strategy for resource r. We already know the solution to subproblem r, so we only need to repeat the above process for any other resource r to obtain the solution $\boldsymbol{X_i}$ of the original problem.

Algorithm 1 ODCA

Input: α_i: a constant; $B_{i,r}$: budget of EU i used for resource r; $S_1 = \{j | j \in \mathscr{M}\}$: set of MECs; $S_z = \{j | p_{j,r} > 0, x_{i,j,r} = 0\} = NULL$: set of MECs that have resource r but are not required.
Output: optimal $x_{i,j,r}, j \in \mathscr{M}$

for (each EU i) **do**
  set $j \leftarrow 1$
  while $j \leq M$ **do**
    if $p_{j,r} = 0$ **then**
      set $x_{i,j,r} \leftarrow 0$
      set $j \leftarrow j + 1$
    else
      if $j \in S_z$ **then**
        set $x_{i,j,r} \leftarrow 0$
        set $j \leftarrow j + 1$
      else
        set $S_{new} \leftarrow S_1 \setminus S_z$, $S' \leftarrow \{j | j \in S_{new}, p_{j,r} = 0\}$
        set $M_{new} \leftarrow |S_{new}| - |S'|$
        $x_{i,j,r} \leftarrow \frac{B_{i,r} + \alpha_i \sum_{j \in S_{new}} p_{j,r}}{M_{new} p_{j,r}} - \alpha_i$
        if $x_{i,j,r} < 0$ **then**
          set $x_{i,j,r} \leftarrow 0$
          set $S_z \leftarrow S_z \cup \{j\}$
          Break;
        else
          set $j \leftarrow j + 1$
        end if
      end if
    end if
  end while
end for
return $x_{i,j,r}, j \in \mathscr{M}$

4.2. MEC problems

Using the given unit price matrix of leader MECs, we analyze how follower EUs select demand strategies. Subsequently, we discuss the leader MECs' utility maximization problem.

Presumably, leader MECs are aware that their actions influence the demand choices of follower EUs. This relationship is summarized by the reaction function $f(\boldsymbol{P})$, which can be obtained through the ODCA. Hence, leader MECs should recognize the influence that they exert on follower EUs when setting the resources' prices.

To obtain maximum utility, MECs should set a suitable price for resources. If MEC j sells resource r at a higher price than other MECs, then the EUs are more willing to obtain resource r from lower-priced MECs. The relationships among MECs are non-cooperative and competitive. We then assume that $Q_{j,r}$ is the amount of resource r in MEC j, $j \in \mathscr{M}$. If MEC j does not have a resource r, then $Q_{j,r} = 0$. We define the utility function for each MEC j as

$$U_j(\boldsymbol{p_j}, \boldsymbol{P_{-j}}) = \sum_{r\in\mathscr{R}} p_{j,r} \sum_{i\in\mathscr{N}} x_{i,j,r}, \tag{27}$$

where $\boldsymbol{P_{-j}}$ is the price matrix of the resources of all other MECs except MEC j. Given $\boldsymbol{P_{-j}}$, MEC j will want to select a proper price vector $\boldsymbol{p_j}$ to maximize its utility. The utility maximization problem for MEC j is expressed as

$$\max_{\boldsymbol{p_j}} \quad U_j(\boldsymbol{p_j}, \boldsymbol{P_{-j}}) \tag{28}$$

$$s.t. \quad \sum_{i\in\mathscr{N}} x_{i,j,r} \le Q_{j,r}, \forall r \in \mathscr{R} \tag{29}$$

$$p_{j,r} \ge 0, \forall r \in \mathscr{R}, \tag{30}$$

where $p_{j,r} = 0$ indicates that MEC j does not contain resource r (i.e., $Q_{j,r} = 0$). EU i's demand for resource r of MEC j is $x_{i,j,r} = 0$. In practice, we forgo this situation when we obtain the optimal price vector $\boldsymbol{p_j}$ through the utility maximization problem. We skip the case where $p_{j,r} = 0$ because the price will never be changed until MEC j newly introduces this resource r. Otherwise, adjusting the price $p_{j,r}$ makes no sense.

Similar to EU optimization problems (Eqs. (2)–(4)), we decompose the utility maximization problem for the MECs (Eqs. (28)–(30)) into R subproblems. We can obtain the optimal price for resource type r in each MEC j through subproblem r.

$$\max_{p_{j,r}} \quad U_{j,r}(p_{j,r}, \boldsymbol{p_{-j,r}}) = p_{j,r} \sum_{i\in\mathscr{N}} x_{i,j,r} \tag{31}$$

$$s.t. \quad \sum_{i\in\mathscr{N}} x_{i,j,r} \le Q_{j,r} \tag{32}$$

$$p_{j,r} \ge 0, j \in \mathscr{M}, \tag{33}$$

where $\boldsymbol{p_{-j,r}}$ is the price vector of resource type r of all other MECs except MEC j and $U_{j,r}$ is the utility of MEC j on resource type r. Constraint (32) captures the capacity limit of resource type r in MEC j. The equality $p_{j,r} = 0$, which is included in constraint (33), occurs if and only if $Q_{j,r} = 0$. This situation is the simplest in the optimization subproblem r. $U_{j,r}$ must be 0 because $p_{j,r} = 0$ and $Q_{j,r} = 0$; maximizing the corresponding utilities does not influence other MECs.

Substituting Eq. (26) into the subproblem r of MEC j (Eqs. (31)–(33)) yields

$$\max_{p_{j,r}} \quad U_{j,r}(p_{j,r}, \boldsymbol{p_{-j,r}}) \tag{34}$$

$$s.t. \quad \sum_{i\in\mathscr{N}/\{i\}:x_{i,j,r}=0} (\frac{B_{i,r} + \alpha_i \sum_{t\in T} p_{t,r}}{|T|\, p_{j,r}} - \alpha_i) \le Q_{j,r} \tag{35}$$

$$p_{j,r} \ge 0, j \in \mathscr{M}, \tag{36}$$

where $U_{j,r}(p_{j,r}, \boldsymbol{p_{-j,r}}) = p_{j,r} \sum_{i\in\mathscr{N}/\{i\}:x_{i,j,r}=0} (\frac{B_{i,r}+\alpha_i \sum_{t\in T} p_{t,r}}{|T| p_{j,r}} - \alpha_i)$.

From the objective function $U_{j,r}(p_{j,r}, \boldsymbol{p_{-j,r}})$, the optimal price $p_{j,r}$ of MEC j is related to the prices of other MECs. Hence, this price optimization scenario can be modeled by the following noncooperative game.

• Players: M MECs.

• Strategies: Each MEC j selects price $p_{j,r} \in C_{j,r}$ to maximize its utility, where $C_{j,r}$ is the set of all values that satisfy constraints (35) and (36).

• Payoffs: The utility function $U_{j,r}$ is defined above. Revenue is the product of the unit price and the number of sold resources.

MECs select price strategies to maximize revenues according to the definitions of the revenues and strategies in the price optimization game. To proceed, the price optimization game should be denoted as $\Gamma = (\mathscr{M}, \{C_{j,r}\}_{j\in\mathscr{M}}, \{U_{j,r}\}_{j\in\mathscr{M}})$ for convenience. The concept of the Nash equilibrium is then introduced [50].

Definition 1. A strategy profile $\boldsymbol{p_r^*} = (p_{1,r}^*, \dots, p_{N,r}^*)$ is the Nash equilibrium of game Γ if no player can further increase its revenue by unilaterally changing its strategy at equilibrium $\boldsymbol{p_r^*}$. Mathematically,

$$U_{j,r}(p_{j,r}^*, \boldsymbol{p_{-j,r}^*}) \ge U_{j,r}(p_{j,r}, \boldsymbol{p_{-j,r}^*}), \forall p_{j,r} \in C_{j,r}, j \in \mathscr{M}. \tag{37}$$

The Nash equilibrium has a satisfactory self-stabilizing property that allows the users at the equilibrium to achieve a mutually satisfactory solution; no user has the incentive to deviate. This property is crucial to the price optimization problem because EUs are different individuals who may act according to their interests.

4.3. Stackelberg game equilibrium analysis

We first study the existence of the Nash equilibrium of the price optimization game among MECs, and we reference work in the literature [7] and [39] to prove the existence of Nash equilibrium.

Theorem 1. *The Nash equilibrium of the price optimization game among MECs always exists and is unique.*

Proof. (1) In game Γ, all players' strategy space is the product space $C_r = C_{1,r} \times C_{2,r} \times \cdots \times C_{N,r}$, where $p_{j,r} \in C_{j,r}, j \in \mathscr{M}$. Thus, C_r is a convex, closed, and non-empty subset of a certain Euclidian space R^M.

(2) The revenue function $U_{j,r}$ is continuous in $p_{j,r}$. The second derivative of $U_{j,r}$ with respect to $p_{j,r}$ is obtained as

$$\frac{\partial^2 U_{j,r}}{\partial p_{j,r}^2} = 0, \forall j \in \mathscr{M}. \tag{38}$$

Therefore, $U_{j,r}$ is concave in $p_{j,r}$ for each fixed value of $\boldsymbol{p_{-j,r}}$, and game Γ is a concave N-person game. In this situation, the Nash equilibrium exists in the game based on Theorem 1 in [51]. Moreover, the Nash equilibrium is unique according to Theorem 3 in [51]. □

We can also prove that the Nash equilibrium of the price optimization game among MECs exists for each resource r. The utility maximization problem in Eqs. (28)–(30) has a Nash equilibrium $\boldsymbol{p_j}$.

EUs determine the amount of each resource purchased from each MEC, whereas MECs set the unit price of each resource. Therefore, we model our problem as a Stackelberg game. The Stackelberg game, which is known as a leader–follower game, studies the decision-making processes of a number of independent players.

MECs, which act as the leaders, determine the price strategy $\boldsymbol{P}$ in the first stage, whereas each EU decides the demands $\boldsymbol{X_i}$ in the second stage. One optimal solution of this Stackelberg game is to reach the Stackelberg equilibrium, which is defined as follows.

Definition 2. Set $(\boldsymbol{P}^*, \boldsymbol{X}^*)$ is called a Stackelberg equilibrium of the game between the MECs and the EUs if it satisfies the following condition.

$U_j(\boldsymbol{P}^*; \boldsymbol{X}^* = f(\boldsymbol{P}^*)) \geq U_j(\boldsymbol{p_j}, \boldsymbol{P}^*_{\boldsymbol{-j}}; \boldsymbol{X} = f(\boldsymbol{p_j}, \boldsymbol{P}^*_{\boldsymbol{-j}})), \forall j \in \mathscr{M}$;
$U_i(\boldsymbol{P}^*; \boldsymbol{X}^* = f(\boldsymbol{P}^*)) \geq U_i(\boldsymbol{P}^*; \boldsymbol{X_i}, \boldsymbol{X}^*_{\boldsymbol{-i}}), \forall i \in \mathscr{N}$.

Theorem 2. *There exists a unique Stackelberg equilibrium for the proposed Stackelberg game.*

Proof. In this Stackelberg game, MECs will select price strategies in the first stage and announce the selected ones to the EUs. According to Theorem 1, a Nash equilibrium solution exists for the price-making process among MECs. In the second stage, EUs will decide the demands for resources based not only on their budgets, but also on the price of each resource in each MEC. EUs can obtain the optimal demand allocation through the proposed ODCA. Therefore, a unique Stackelberg equilibrium exists for the proposed Stackelberg game. □

4.4. Algorithm of reaching equilibrium

Algorithm 2 Algorithm of reaching equilibrium

1: **for** (each resource type r, $r \in \mathscr{R}$) **do**
2:   **for** (each iteration t) **do**
3:     MECs publish the current price of resource type r, $\{p_{j,r}, \forall j \in \mathscr{M}\}$
4:     Each EU i computes the best demand from MEC j $x_{i,j,r}$ for resource r using the proposed ODCA algorithm and sends it to MEC j, where $j \in \mathscr{M}$.
5:     **for** (each MEC j, $j \in \mathscr{M}$) **do**
6:       **if** ($\sum_{i=1}^{N} x_{i,j,r} > Q_{j,r}$) **then**
7:         Increase the resource price of MEC j $p'_{j,r}$.
8:         Update the demands from all MECs.
9:       **else**
10:         **if** ($\sum_{i=1}^{N} x_{i,j,r} < Q_{j,r}$) **then**
11:           Decrease the resource price of MEC j $p'_{j,r}$.
12:           Update the demands from all MECs.
13:         **else**
14:           Price $p_{j,r}$ remains unchanged.
15:         **end if**
16:       **end if**
17:     **end for**
18:     **if** ($|p'_{j,r} - p_{j,r}| < \epsilon, \forall j \in \mathscr{M}$) **then**
19:       $p_{j,r}* \leftarrow p'_{j,r}$;
20:       Return $p_{j,r}*$;
21:       Break;
22:     **else**
23:       t=t+1; Break;
24:     **end if**
25:   **end for**
26: **end for**

In this section, we develop an iterative algorithm to achieve the equilibrium of the proposed Stackelberg game.

Algorithm 2 runs iteratively for each resource type r (lines 2–25). The initial price of resource r $p_{j,r} > 0, \forall j \in \mathscr{M}$ is arbitrarily set by each MEC. Each MEC updates its unit price in each iteration once based on EUs' demands and total resources (lines 2–25). In this algorithm, the mechanism for updating the prices is mathematically expressed as

$$p'_{j,r} = p_{j,r} + (\textstyle\sum_{i=1}^{N} x_{i,j,r} - Q_{j,r}) \times \delta_j, \tag{39}$$

where δ_j is a step parameter of MEC j, which is a sufficiently small number. Once an MEC updates its prices, the EUs will adjust their demands based on the current situation by using our proposed ODCA (line 4).

The unit prices of the MECs and the demands of EUs update interactively. The algorithm terminates when the price profiles obtained by two successive iterations are close enough and the accuracy requirement is satisfied (lines 18–21). The final converged price profile $\boldsymbol{p_r}* = (p_{1,r}*, p_{2,r}*, \ldots, p_{M,r}*)$ and the corresponding EUs' price profiles construct the Stackelberg equilibrium for resource type r. A Stackelberg equilibrium solution for the proposed Stackelberg game can be obtained by repeating the aforementioned process for other resources.

5. Experimental evaluation

In this section, we illustrate the performance of the proposed Stackelberg Game Approach (SGA).

Research on mobile edge computing is still at infancy. To our knowledge, few studies have tackled the allocation and pricing of resources [3,28,52]. However, because of the different background information and optimization objectives, comparing SGA with the previous works is not appropriate. In accordance to the methods in [7,15,53,54], and [36], we design several numerical examples to evaluate the performances of the proposed SGA. We investigate how EUs select the best resource combinations based on the unit price of each resource in each MEC under limited budgets. We also study how MECs determine the unit price of the resources to maximize revenues. Without loss of generality, we also consider the equilibrium of the Stackelberg game for the scenario comprising a large number of MECs and EUs. Then, we show the convergence of Algorithm 2 and how an EU with idle resources can play the role of an MEC. Moreover, because we apply a Stackelberg game to model the interaction between MECs and EUs in a decentralized manner, we consider the following three centralized schemes as baseline schemes for comparison.

- Interior Point Algorithm (IPA) [55,56]
- Sequential Quadratic Programming algorithm (SQP) [57,58]
- Active Set Algorithm (ASA) [59,60]

The objective of the three centralized schemes is to maximize the sum of the utility of all individuals, including MECs and EUs.

Unless stated otherwise, the parameters used in the analysis are as follows: $\alpha_i = 1$; $N = 5$ EUs; $M = 3$ MECs; $R = 3$ resources in the mobile edge computing system; budget limits of EUs are $B_{1,1} = 5$, $B_{1,2} = 15$, $B_{1,3} = 10$; $B_{2,1} = 7$, $B_{2,2} = 16$, $B_{2,3} = 15$; $B_{3,1} = 9$, $B_{3,2} = 4$, $B_{3,3} = 20$; $B_{4,1} = 12$, $B_{4,2} = 10$, $B_{4,3} = 25$; $B_{5,1} = 15$, $B_{5,2} = 9$, and $B_{5,3} = 30$; $\alpha_i = 1$ for all $i \in \mathscr{N}$; the available resources of MECs include $Q_{1,1} = 10$, $Q_{1,2} = 11$, $Q_{1,3} = 30$; $Q_{2,1} = 15$, $Q_{2,2} = 27$, $Q_{2,3} = 30$; $Q_{3,1} = 20$, $Q_{3,2} = 26$, and $Q_{3,3} = 30$; $\delta_j = 0.01$; and $\epsilon = 10^{-10}$.

5.1. Simulation steps

MATLAB is used to examine the performance of the proposed Stackelberg game. The simulation steps are performed using iterations to obtain the Stackelberg equilibrium. We take a single resource type as an example. First, given the unit price of the resource of each MEC, each EU makes the best choice for the resource according to the proposed ODCA. Then, the MECs will update their prices based on the EUs' requirements. This process is repeated until the accuracy requirement for the unit price is satisfied. Finally, when the unit prices are determined, each EU's optimal resource demand can be obtained through the ODCA.

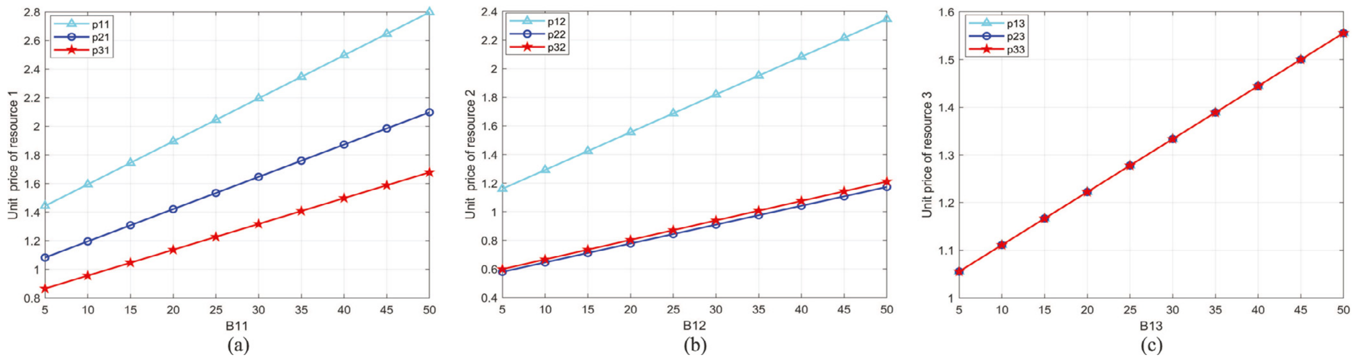


Fig. 2. Optimal prices of MECs at the Stackelberg equilibrium. (a) Unit price of resource 1 vs. B_{11}. (b) Unit price of resource 2 vs. B_{12}. (c) Unit price of resource 3 vs. B_{13}.

Table 3
Unit price of resource r at the Stackelberg equilibrium when $N = 100$ EUs and $M = 7$ MECs with a variation in $B_{1,r}$.

$B_{1,r}$	$p_{1,r}$	$p_{2,r}$	$p_{3,r}$	$p_{4,r}$	$p_{5,r}$	$p_{6,r}$	$p_{7,r}$
10	5.0791	3.7630	3.0104	2.5087	2.1503	1.8815	1.8815
20	5.0894	3.7749	3.0199	2.5166	2.1571	1.8874	1.8874
30	5.1058	3.7864	3.0291	2.5243	2.1637	1.8932	1.8932
40	5.1223	3.7979	3.0384	2.5320	2.1703	1.8990	1.8990
50	5.1388	3.8095	3.0476	2.5396	2.1768	1.9047	1.9047
60	5.1553	3.8210	3.0568	2.5473	2.1834	1.9105	1.9105
70	5.1717	3.8325	3.0660	2.5550	2.1900	1.9163	1.9163
80	5.1882	3.8440	3.0752	2.5627	2.1966	1.9220	1.9220
90	5.2047	3.8556	3.0845	2.5704	2.2032	1.9278	1.9278
100	5.2211	3.8671	3.0937	2.5781	2.2098	1.9335	1.9335

5.2. Example 1: Optimal prices and utilities of MECs at the Stackelberg equilibrium

Fig. 2 shows the optimal prices determined by each MEC at the Stackelberg equilibrium when the budget of EU 1 for each kind of resource varies from 5 to 50. Fig. 2(a) and (b) show that the price for an MEC with few resources is higher than that with numerous resources. For example, MEC 1 charges higher than MEC 2 for Resource 1, and MEC 2 charges higher than MEC 3, which is because the amount of Resource 1 that the three MECs possess is 10, 15, and 20, respectively. When the three MECs have the same amount of resources, EUs cannot distinguish among them, and the charges are the same (Fig. 2(c)). Fig. 2 shows that the increases in the unit price of each resource is roughly linear as the budget of the EUs increases.

Fig. 3(a), (b), and (c) show the revenue of each MEC with one resource type, which corresponds to the situations in Fig. 2(a), (b), and (c), respectively. Fig. 3(d) depicts the total revenue of each MEC when the total budget of EU 1 varies from 15 to 150. Although MEC 1 charges the highest among the MECs, the corresponding revenue is the lowest because the MEC possesses the lowest amount resources (Fig. 3(a) and (b)). By contrast, the revenue of MEC 3 is the highest despite having the lowest unit price among the MECs (Figs. 2(a) and 3(a)).

5.3. Example 2: Optimal demands and utilities of EUs at the Stackelberg equilibrium

Fig. 4(a), (b), and (c) display the demands of each EU for each resource at the Stackelberg equilibrium. EU 1's demand for resources increases with the increase in the corresponding budget. On the contrary, the demands of other EUs decrease because the price of the resources increased due to the increment in the budget of EU 1 while their budget remains unchanged. Fig. 5 shows the utilities of the EUs at the Stackelberg equilibrium.

For a certain resource r, we consider the equilibrium of the Stackelberg game for large-scale applications consisting of a large number of MECs and EUs. Tables 3 and 4 summarize the results of the unit price and EU demands at the Stackelberg equilibrium when the number of EUs and MECs are 100 and 7, respectively, and the budget of EU 1 is $B_{1,r} = 10, 20, \ldots, 100$. The budgets of EUs 2–10, 11–20, 21–30, 31–40, 41–50, 51–60, 61–70, 71–80, 81–90, and 91–100 are 10, 15, 20, 25, 30, 35, 40, 45, 50, and 55, respectively. The available resource r of the MECs are $Q_{1,r} = 50$, $Q_{2,r} = 100$, $Q_{3,r} = 150$, $Q_{4,r} = 200$, $Q_{5,r} = 250$, and $Q_{6,r} = Q_{7,r} = 300$. The characteristics included in Tables 3 and 4 are similar to those in Figs. 2 and 4.

5.4. Example 3: EU with idle resources acting as an MEC

In a mobile edge computing system, an EU with idle resources can act as an MEC. For example, for several kinds of resources, if few EUs own resources with idle states, then the EUs can serve as MECs to provide the resources to other requiring EUs. Furthermore, the EUs can charge other EUs for providing the resources. Taking resource 1 as an example, suppose that an EU with available resource 1 exists, and the amount of resource 1 is $Q_{EU,1} = 5$ (Fig. 6(a) and (b)). The parameters of the other MECs

Table 4
EU demands for resource r at the Stackelberg equilibrium when $N = 100$ EUs and $M = 7$ MECs with a variation in $B_{1,r}$.

$B_{1,r}$	$EU1$	$EU2-10$	$EU11-20$	$EU21-30$	$EU31-40$	$EU41-50$	$EU51-60$	$EU61-70$	$EU71-80$	$EU81-90$	$EU91-100$
10	4.6013	4.6013	6.7051	8.6580	10.6019	12.5458	14.4897	16.4337	18.3776	20.3215	22.2654
20	8.6325	4.5880	6.6852	8.6325	10.5704	12.5084	14.4463	16.3843	18.3222	20.2601	22.1981
30	12.4731	4.5753	6.6661	8.6091	10.5411	12.4731	14.4051	16.3371	18.2691	20.2012	22.1332
40	16.2903	4.5626	6.6470	8.5858	10.5119	12.4380	14.3641	16.2903	18.2164	20.1425	22.0687
50	20.0843	4.5500	6.6281	8.5626	10.4829	12.4031	14.3234	16.2437	18.1640	20.0843	22.0045
60	23.8552	4.5374	6.6093	8.5396	10.4540	12.3685	14.2830	16.1974	18.1119	20.0263	21.9408
70	27.6035	4.5250	6.5906	8.5167	10.4254	12.3341	14.2427	16.1514	18.0601	19.9688	21.8774
80	31.3291	4.5126	6.5720	8.4940	10.3969	12.2998	14.2028	16.1057	18.0086	19.9115	21.8145
90	35.0324	4.5003	6.5536	8.4714	10.3686	12.2658	14.1630	16.0602	17.9574	19.8547	21.7519
100	38.7135	4.4880	6.5352	8.4489	10.3404	12.2320	14.1235	16.0150	17.9066	19.7981	21.6897

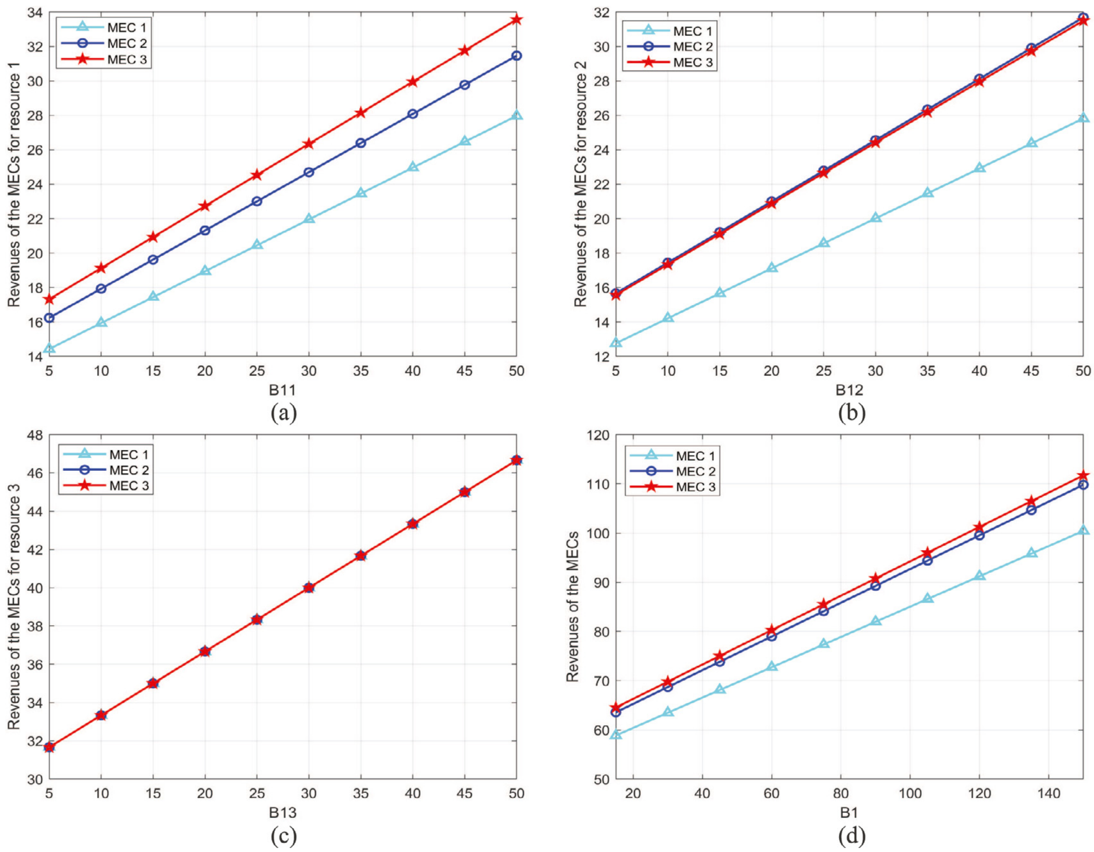


Fig. 3. Optimal utilities of MECs at the Stackelberg equilibrium. (a) Revenues of the MECs for resource 1 vs. B_{11}. (b) Revenues of the MECs for resource 2 vs. B_{12}. (c) Revenues of the MECs for resource 3 vs. B_{13}. (d) Revenues of the MECs vs. B_1.

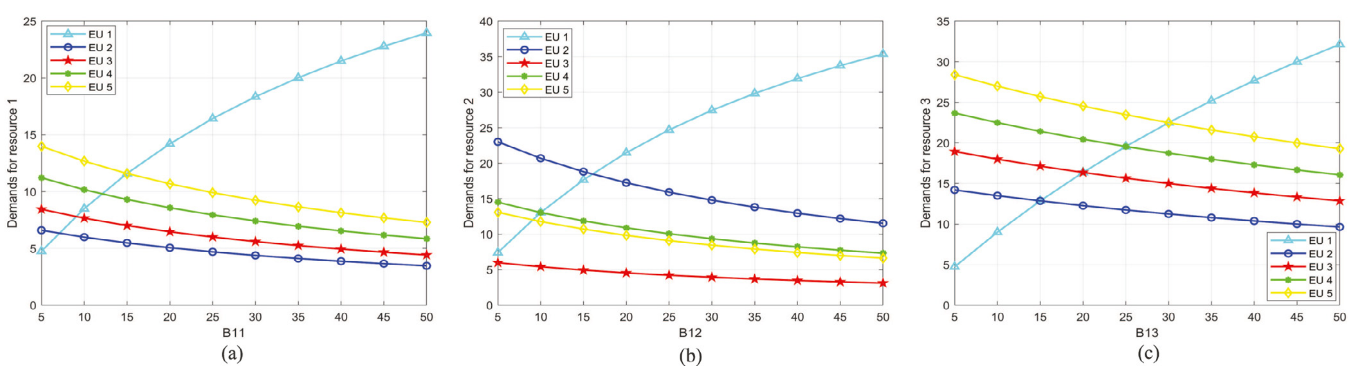


Fig. 4. EUs' demands at the Stackelberg equilibrium. (a) EUs' demands for resource 1 vs. B_{11}. (b) EUs' demands for resource 2 vs. B_{12}. (c) EUs' demands for resource 3 vs. B_{13}.

are the same as the parameters set at the beginning of Section 5. The main observations from Fig. 6 are elaborated as follows.

(1) The unit price of the original MECs decreases due to the increase in resources. Another reason is the introduction of a newly joined MEC that brings this kind of resource. Moreover, the price of the newly joined MEC is higher than any other original MEC because the amount of resources in the former is the minimum quantity.

(2) The demands for Resource 1 of all EUs increase due to the decrease in the prices of the originally existing MECs and the unchanged budgets of EUs.

(3) The newly joined MEC can obtain revenues because of its ability to provide its own idle resources for the EUs.

Therefore, any EU with idle resources can act as an MEC.

5.5. Example 4: Convergence of Algorithm 2

Three parameters affect the convergence speed of Algorithm 2: the initial price, accuracy requirement ϵ, and step parameter δ_j. The pricing problem of Resource 1 will be treated as an example to analyze the effect of these parameters on Algorithm 2.

(1) Initial unit price. We set the accuracy requirement $\epsilon = 10^{-10}$ and $\delta_j = 0.05$. Fig. 7 shows that different initial prices exert an effect on the convergence speed of the algorithm. The closer the initial unit price is to the equilibrium price, the faster Algorithm 2 converges.

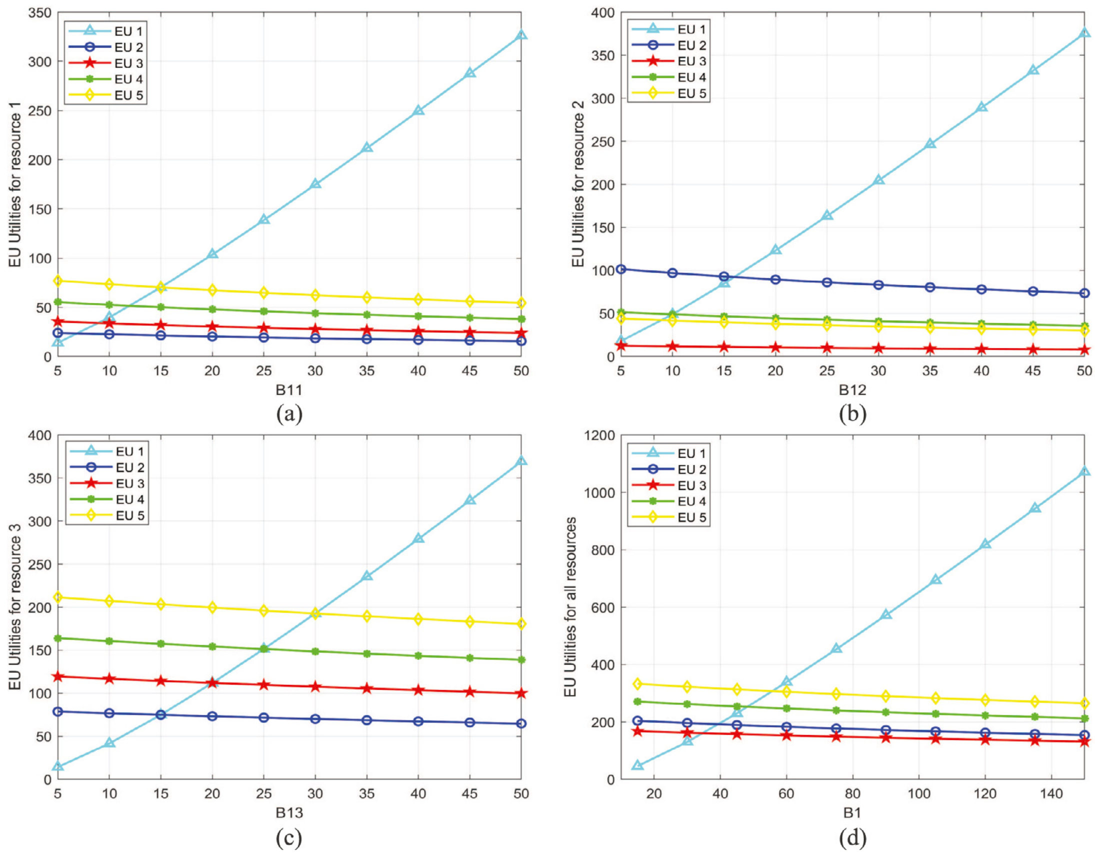


Fig. 5. EU utilities at the Stackelberg equilibrium. (a) EU utilities for resource 1 vs. B_{11}. (b) EU utilities for resource 2 vs. B_{12}. (c) EU utilities for resource 3 vs. B_{13}. (d) EU utilities vs. B_1.

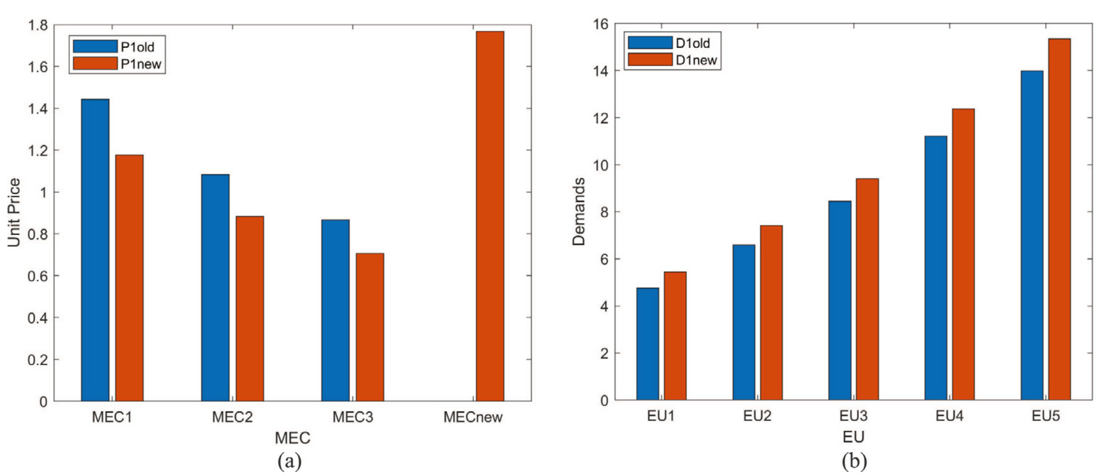


Fig. 6. Changes in MECs and EUs when an EU with idle resources acts as an MEC. (a) Changes in the unit price of MECs' resources. (b) Changes in EUs' demands.

(2) ϵ. To eliminate the influence of the other two parameters, we set $\delta_j = 0.02$ and $j = 1, 2, 3$. The experiment is repeated 100 times to eliminate the effect of the randomness of the initial price. The average results are summarized in Table 5, which also displays the number of iterations for $\epsilon = 10^{-1}, 10^{-2}, \ldots, 10^{-10}$. The higher the accuracy requirement (i.e., the lower the value of ϵ), the higher the number of iterations t and the closer the unit price is to the optimal one.

(3) δ_j. We set the initial price of MECs as $p_{1,1} = p_{2,1} = p_{3,1} = 6$ and $\epsilon = 10^{-10}$. Fig. 8(a) and (b) indicate that Algorithm 2 converges faster when the step parameter δ_j is larger.

5.6. Example 5: Decentralized mechanism: SGA vs. centralized schemes

To compare and assess the performance of the proposed decentralized mechanism (SGA), we implement the system-wide utility maximization solution (i.e., $\max_{X,P} \sum_{i\in\mathcal{N},j\in\mathcal{M}} U_i + U_j$) using three centralized optimization schemes, namely, IPA, SQP, and ASA.

For a certain resource type 1, we run several experiments to compare the system-wide total utility when the budget of EU 1 is $B_{1,1} = 5, 10, \ldots, 50$. The other parameters of MECs and

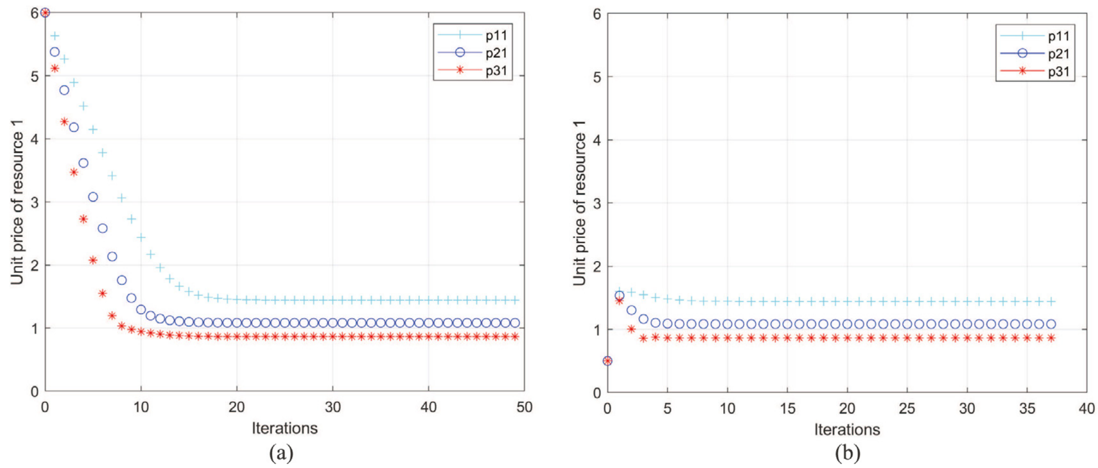


Fig. 7. Unit price of MECs' resource 1 with different initial unit prices. (a) $p_{1,1} = p_{2,1} = p_{3,1} = 6$. (b) $p_{1,1} = p_{2,1} = p_{3,1} = 0.5$.

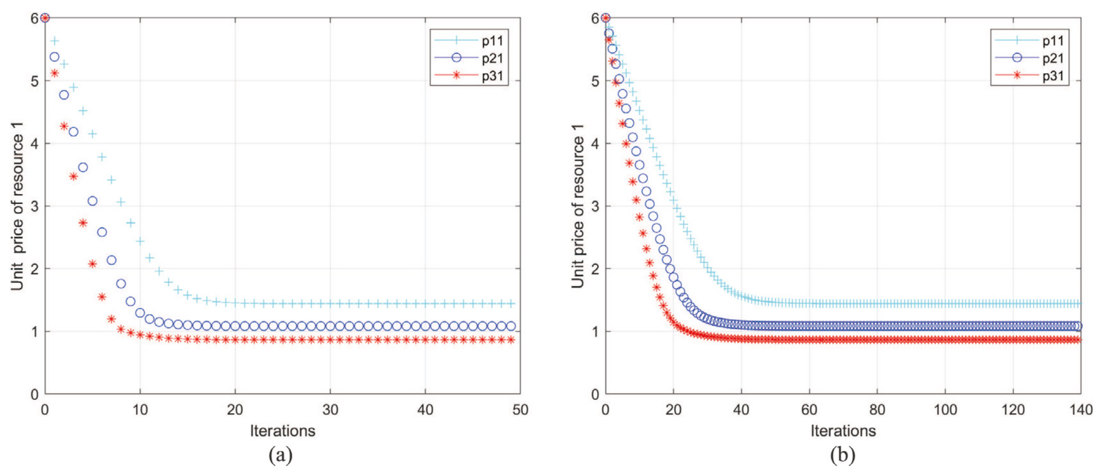


Fig. 8. Unit price of MECs' resource 1 with different values of δ_j, $(j = 1, 2, 3)$. (a) $\delta_j = 0.05$. (b) $\delta_j = 0.02$.

Table 5
Unit price of MECs' resource 1 with different ϵ.

Accuracy requirement ϵ	t	p_{11}	p_{21}	p_{31}
10^{-1}	26	1.9299	1.2985	0.9544
10^{-2}	47	1.4861	1.0921	0.8708
10^{-3}	52	1.4477	1.0835	0.8666
10^{-4}	67	1.4440	1.0828	0.8662
10^{-5}	80	1.4437	1.0827	0.8662
10^{-6}	92	1.4436	1.0827	0.8662
10^{-7}	109	1.4436	1.0827	0.8662
10^{-8}	117	1.4436	1.0827	0.8662
10^{-9}	127	1.4436	1.0827	0.8662
10^{-10}	140	1.4436	1.0827	0.8662

Table 6
Comparison of the total utilities with a variation in $B_{1,1}$.

$B_{1,1}$	SGA	IPA	SQP	ASA
5	253.7504	254.2595	254.0912	254.3389
10	274.8481	275.0994	275.0144	275.0994
15	301.8510	302.2093	302.0904	302.2093
20	332.4434	333.0566	332.8519	333.0566
25	365.5030	366.4621	366.1398	366.4621
30	400.3798	401.7537	401.2905	401.2905
35	436.6574	438.5034	437.8808	437.8808
40	474.0498	476.4186	475.6210	476.7900
45	512.3516	515.2890	514.3026	515.7489
50	551.4092	554.9580	554.9580	555.5123

EUs are the same as the parameters set at the beginning of the experimental section. The results are presented in Table 6.

The findings from Table 6 indicate that the system-wide total utility of all mechanisms increases as the budget of EU 1 increases. The total utility of SGA is slightly lower than that of the three centralized methods. Compared with the centralized optimization solution, the total utility obtained by SGA is almost equal to that obtained by IPA, SQP, and ASA, and the performance loss of the former is lower by 1% in all cases. This result illustrates the feasibility and effectiveness of the proposed decentralized mechanism in terms of total utility.

Taking resource 1 again as an example, we investigate the comparison of the utility of each individual method after the allocation and pricing of resource 1 is determined through SGA and the three centralized mechanisms. The parameters are the same as the parameters set at the beginning of the experiments. Fig. 9 describes the utility values of the three MECs and five UEs obtained by different algorithms. The utility obtained by EUs 1, 2, and EU 3 using SGA is the largest among the compared algorithms. The utilities obtained by EUs 4 and 5 through SGA are not the largest, but are extremely close to the maximum value. MEC 3 only obtained an effective utility value through SGA; the utility

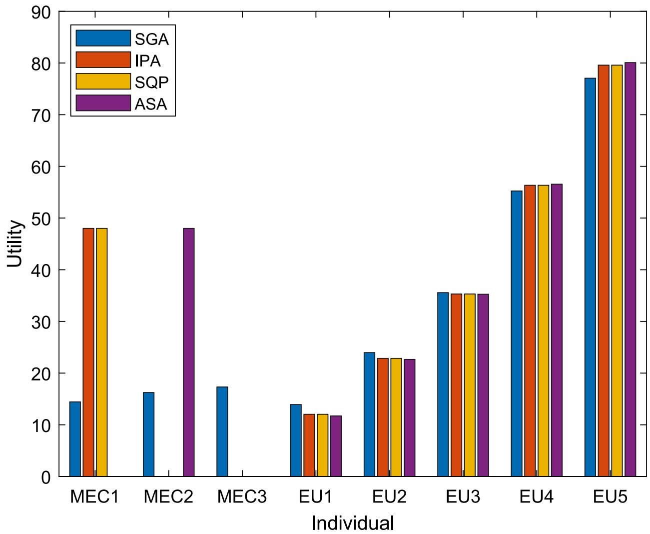


Fig. 9. Individual utility comparison.

value is equal to 0 when the three methods are employed. For MEC 1, the utility value obtained by the ASA method is 0, whereas for MEC 2, the utility value obtained through this method is the largest. MEC 1 obtained a large utility value through IPA and SQP methods. However, that of MEC 2 is 0, which indicates that the individual utility induced by IPA, SQP, and ASA is unbalanced. By contrast, SGA delivered satisfactory utility values.

5.7. Example 6: Sensitivity analysis of the parameters affecting the proposed SGA

To investigate the sensitivity analysis of the parameters that mainly affect the proposed SGA, we consider two different scenarios, in which resource r is treated as an example in both scenarios. In the first scenario, we analyze the effect of the different numbers of EUs on the unit price of resource r in each MEC and on the corresponding revenues. We assume three MECs with $Q_{1,r} = 10$, $Q_{2,r} = 15$ and $Q_{3,r} = 20$, and the number of EUs is within 4–20. To simplify the problem, we assume that each EU's budget is 1. In the other scenario, we consider the effect of different numbers of MECs on the demands of EUs for resource r and on the corresponding utilities. The number of MECs varies from 3 to 15. The number of resource r in each MEC is set to 100, and the budgets of EUs 1–20, 21–40, 41–60, 61–80, and 81–100 are set to 10, 20, 30, 40, and 50, respectively.

Fig. 10(a) shows that with the increase in EUs, the total budget of all EUs for resource r increases, resulting in the increase in the unit price of resource r of each MEC. Correspondingly, as the number of EUs and the unit price of resource r increase, the revenue of each MEC increases (Fig. 10(b)). Although the budgets of EUs for resource r are the same in all cases (i.e., all budgets are set to 1), the amount of resources that each EU can obtain decreases because the unit price increases while the budget for resource r remains unchanged. Moreover, the total amount of resource r does not change, but the number of EUs demanding resource r increases. This situation is consistent with the market law of under-supply in economics.

Fig. 11(a) and (b) presents the demands and utilities for resource r for 100 EUs. The number of MECs varies from 3 to

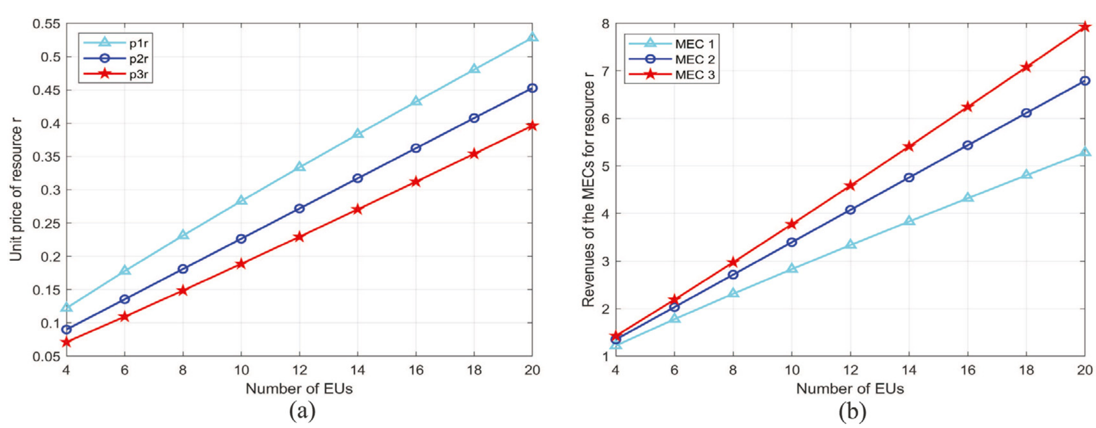


Fig. 10. Equilibrium for MECs under various EUs. (a) Unit price of resource r. (b) Revenues of the MECs for resource r.

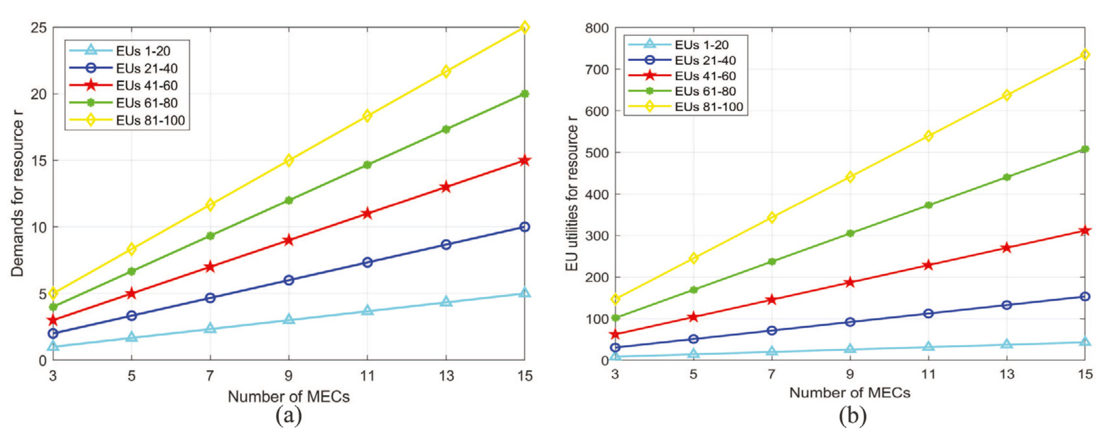


Fig. 11. Equilibrium for EUs under various MECs. (a) Demands for resource r. (b) EU utilities for resource r.

15. The total amount of resource r increases as the number of MECs grows. Therefore, each EU demands increasing resources (Fig. 11(a)). Similarly, the utilities of all EUs increase due to the increase in the demands. Although the budgets of the EUs for resource r do not change in all cases, the demands for resource r increase because the total number of resource r increases while the number of EUs remains the same. This situation is consistent with the market law of over-supply in economics.

6. Conclusion

In this paper, we study multiple resources allocation and pricing issues in a mobile edge computing system and propose a Stackelberg game-based framework in which the EUs act as followers and MECs act as leaders. To solve this problem efficiently, we decompose the multiple resources allocation and pricing problem into a set of subproblems, each of which only considers a single resource type. For each subproblem, we first prove the existence of the Stackelberg equilibrium of the game. To determine the Stackelberg equilibrium for each resource type, we then develop an algorithm ODCA to find the best demand strategy of the resource for an EU and an iterative algorithm to find an equilibrium price. The numerical results demonstrate that the proposed mechanism is efficient and scales well as the system size increases. Moreover, we show that an EU with idle resources can play the role of an MEC.

In future work, we will consider a dynamic scenario wherein EUs may depart and leave within a computation offloading period. In addition, we will conduct a research that not only combines resource allocation with specific task models that involve transmissions over wireless channels but also accounts for the energy efficiency of EUs.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Yifan Chen: Data curation, Formal analysis, Investigation, Methodology, Validation, Writing - original draft, Writing - review & editing. **Zhiyong Li:** Conceptualization, Project administration, Funding acquisition. **Bo Yang:** Formal analysis, Methodology, Writing - review & editing. **Ke Nai:** Methodology, Writing - review & editing. **Keqin Li:** Conceptualization.

Acknowledgment

This work was partially supported by National Key Research and Development Program of China (No. 2018YFB1308604), National Natural Science Foundation of China (No. 61672215, No. 61976086), Hunan Science and Technology Innovation Project (No. 2017XK2102) and Hunan Innovation Technology Investment Project (No. 2019GK5061). The correspondence author is Zhiyong Li.

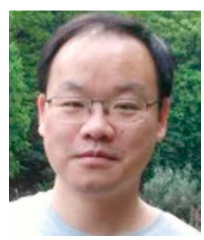
References

[1] W. Shi, J. Cao, Q. Zhang, Y. Li, L. Xu, Edge computing: Vision and challenges, IEEE Internet Things J. 3 (5) (2016) 637–646.
[2] M. Chiang, T. Zhang, Fog and iot: An overview of research opportunities, IEEE Internet Things J. 3 (6) (2016) 854–864.
[3] D.T. Nguyen, L.B. Le, V.K. Bhargava, Price-based resource allocation for edge computing: A market equilibrium approach, IEEE Trans. Cloud Comput. (2018) 1.
[4] R. Roman, J. Lopez, M. Mambo, Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges, Future Gener. Comput. Syst. 78 (1) (2018) 680–698.
[5] P. Mach, Z. Becvar, Mobile edge computing: A survey on architecture and computation offloading, IEEE Commun. Surv. Tutor. PP (99) (2017) 1.
[6] Y. Mao, C. You, J. Zhang, K. Huang, K.B. Letaief, A survey on mobile edge computing: The communication perspective, IEEE Commun. Surv. Tutor. PP (99) (2017) 1.
[7] K. Li, A game theoretic approach to computation offloading strategy optimization for non-cooperative users in mobile edge computing, IEEE Trans. Sustain. Comput. (2018) 1, http://dx.doi.org/10.1109/TSUSC.2018.2868655.
[8] W. Chen, D. Wang, K. Li, Multi-user multi-task computation offloading in green mobile edge cloud computing, IEEE Trans. Serv. Comput. (2018) 1, http://dx.doi.org/10.1109/TSC.2018.2826544.
[9] Y. Mao, J. Zhang, K.B. Letaief, Dynamic computation offloading for mobile-edge computing with energy harvesting devices, IEEE J. Sel. Areas Commun. 34 (12) (2016) 3590–3605.
[10] L. Yang, H. Zhang, M. Li, J. Guo, H. Ji, Mobile edge computing empowered energy efficient task offloading in 5g, IEEE Trans. Veh. Technol. 67 (7) (2018) 6398–6409, http://dx.doi.org/10.1109/TVT.2018.2799620.
[11] N.C. Luong, P. Wang, D. Niyato, Y. Wen, Z. Han, Resource management in cloud networking using economic analysis and pricing models: A survey, IEEE Commun. Surv. Tutor. 19 (2) (2017) 954–1001.
[12] S. Penmatsa, A.T. Chronopoulos, Cost minimization in utility computing systems, Concurr. Comput.: Pract. Exper. 26 (1) (2014) 287–307.
[13] S. Penmatsa, A.T. Chronopoulos, Game-theoretic static load balancing for distributed systems, J. Parallel Distrib. Comput. 71 (4) (2011) 537–555.
[14] D. Ardagna, M. Ciavotta, M. Passacantando, Generalized nash equilibria for the service provisioning problem in multi-cloud systems, IEEE Trans. Serv. Comput. 10 (3) (2017) 381–395.
[15] C. Liu, K. Li, K. Li, A game approach to multi-servers load balancing with load-dependent server availability consideration, IEEE Trans. Cloud Comput. (2018) 1, http://dx.doi.org/10.1109/TCC.2018.2790404.
[16] J. Hu, K. Li, C. Liu, K. Li, A game-based price bidding algorithm for multi-attribute cloud resource provision, IEEE Trans. Serv. Comput. (2018) 1, http://dx.doi.org/10.1109/TSC.2018.2860022.
[17] B. Yang, Z. Li, S. Jiang, K. Li, Envy-free auction mechanism for vm pricing and allocation in clouds, Future Gener. Comput. Syst. 86 (2018) 680–693.
[18] M.M. Nejad, L. Mashayekhy, D. Grosu, Truthful greedy mechanisms for dynamic virtual machine provisioning and allocation in clouds, IEEE Trans. Parallel Distrib. Syst. 26 (2) (2014) 594–603.
[19] G.D. Modica, O. Tomarchio, P. Bonacquisto, G. Petralia, A procurement auction market to trade residual cloud computing capacity, IEEE Trans. Cloud Comput. 3 (3) (2015) 345–357.
[20] M. Tanaka, Y. Murakami, Strategy-proof pricing for cloud service composition, IEEE Trans. Cloud Comput. 4 (3) (2016) 363–375.
[21] Y. Mao, J. Zhang, S. Song, K.B. Letaief, Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems, IEEE Trans. Wireless Commun. 16 (9) (2017) 5994–6009.
[22] C. You, K. Huang, H. Chae, B.-H. Kim, Energy-efficient resource allocation for mobile-edge computation offloading, IEEE Trans. Wireless Commun. 16 (3) (2017) 1397–1411.
[23] J. Guo, Z. Song, Y. Cui, Z. Liu, Y. Ji, Energy-efficient resource allocation for multi-user mobile edge computing, in: GLOBECOM 2017-2017 IEEE Global Communications Conference, IEEE, 2017, pp. 1–7.
[24] L. Yin, J. Luo, H. Luo, Tasks scheduling and resource allocation in fog computing based on containers for smart manufacturing, IEEE Trans. Ind. Inf. 14 (10) (2018) 4712–4721.
[25] S. Sardellitti, G. Scutari, S. Barbarossa, Joint optimization of radio and computational resources for multicell mobile-edge computing, IEEE Trans. Signal Inf. Process. Netw. 1 (2) (2015) 89–103.
[26] T.X. Tran, D. Pompili, Joint task offloading and resource allocation for multi-server mobile-edge computing networks, IEEE Trans. Veh. Technol. 68 (1) (2019) 856–868, http://dx.doi.org/10.1109/TVT.2018.2881191.
[27] A.-L. Jin, W. Song, P. Wang, D. Niyato, P. Ju, Auction mechanisms toward efficient resource sharing for cloudlets in mobile cloud computing, IEEE Trans. Serv. Comput. 9 (6) (2016) 895–909.
[28] A.-L. Jin, W. Song, W. Zhuang, Auction-based resource allocation for sharing cloudlets in mobile cloud computing, IEEE Trans. Emerg. Top. Comput. 6 (1) (2018) 45–57.
[29] W. Sun, J. Liu, Y. Yue, H. Zhang, Double auction-based resource allocation for mobile edge computing in industrial internet of things, IEEE Trans. Ind. Inf. 14 (10) (2018) 4692–4701.
[30] D. Gale, L.S. Shapley, College admissions and the stability of marriage, Amer. Math. Monthly 69 (1) (1962) 9–15.

[31] A.E. Roth, M. Sotomayor, Two-sided matching, Handb. Game Theory Econ. Appl. 1 (1992) 485–541.
[32] J. Wang, Y. Long, J. Wang, S.M. Errapotu, H. Li, M. Pan, Z. Han, D-frost: Distributed frequency reuse-based opportunistic spectrum trading via matching with evolving preferences, IEEE Trans. Wireless Commun. 17 (6) (2018) 3794–3806.
[33] O. Semiari, W. Saad, M. Bennis, B. Maham, Caching meets millimeter wave communications for enhanced mobility management in 5g networks, IEEE Trans. Wireless Commun. 17 (2) (2017) 779–793.
[34] H. Zhang, M. Bennis, L.A. DaSilva, Z. Han, Multi-leader multi-follower stackelberg game among wi-fi, small cell and macrocell networks, in: 2014 IEEE Global Communications Conference, IEEE, 2014, pp. 4520–4524.
[35] Y. Wang, X. Lin, M. Pedram, A stackelberg game-based optimization framework of the smart grid with distributed pv power generations and data centers, IEEE Trans. Energy Convers. 29 (4) (2014) 978–987.
[36] H. Zhang, Y. Xiao, S. Bu, D. Niyato, F.R. Yu, Z. Han, Computing resource allocation in three-tier iot fog networks: A joint optimization approach combining stackelberg game and matching, IEEE Internet Things J. 4 (5) (2017) 1204–1215.
[37] V. Cardellini, V.D. Valerio, F.L. Presti, Game-theoretic resource pricing and provisioning strategies in cloud systems, IEEE Trans. Serv. Comput. PP (99) (2016) 1.
[38] B. Yang, Z. Li, S. Chen, T. Wang, K. Li, Stackelberg game approach for energy-aware resource allocation in data centers, IEEE Trans. Parallel Distrib. Syst. 27 (12) (2016) 3646–3658.
[39] S. Maharjan, Q. Zhu, Y. Zhang, S. Gjessing, T. Basar, Dependable demand response management in the smart grid: A stackelberg game approach, IEEE Trans. Smart Grid 4 (1) (2013) 120–132.
[40] X. Kang, R. Zhang, M. Motani, Price-based resource allocation for spectrum-sharing femtocell networks: A stackelberg game approach, IEEE J. Sel. Areas Commun. 30 (3) (2012) 538–549.
[41] S. Guo, X. Hu, G. Dong, W. Li, X. Qiu, Mobile edge computing resource allocation: A joint stackelberg game and matching strategy, Int. J. Distrib. Sens. Netw. 15 (7) (2019) 1550147719861556.
[42] Z. Xiong, J. Kang, D. Niyato, P. Wang, V. Poor, Cloud/edge computing service management in blockchain networks: Multi-leader multi-follower game-based admm for pricing, IEEE Trans. Serv. Comput. (2019).
[43] X. Lin, J. Li, W. Yang, J. Wu, Z. Zong, X. Wang, Vehicle-to-cloudlet: Game-based computation demand response for mobile edge computing through vehicles, in: 2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring), IEEE, 2019, pp. 1–6.
[44] https://amazonaws-china.com/cn/ec2/instance-types/.
[45] H.R. Varian, Intermediate Microeconomics: A Modern Approach: Ninth International Student Edition, WW Norton & Company, 2014.
[46] F.P. Kelly, A.K. Maulloo, D.K.H. Tan, Rate control for communication networks: shadow prices, proportional fairness and stability, J. Oper. Res. Soc. 49 (3) (1998) 237–252.
[47] N. Nisan, T. Roughgarden, E. Tardos, V.V. Vazirani, Algorithmic Game Theory, Cambridge University Press, 2007.
[48] T. Basar, R. Srikant, Revenue-maximizing pricing and capacity expansion in a many-users regime, in: Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, Vol. 1, 2002, pp. 294–301.
[49] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge university press, 2004.
[50] D. Fudenberg, J. Tirole, Game theory, 1991, Camb. Mass. 393 (12) (1991) 80.
[51] J.B. Rosen, Existence and uniqueness of equilibrium points for concave n-person games, Econometrica (1965) 520–534.
[52] D.T. Nguyen, L.B. Le, V. Bhargava, A market-based framework for multi-resource allocation in fog computing, 2018.
[53] C. Liu, K. Li, K. Li, R. Buyya, A new cloud service mechanism for profit optimizations of a cloud provider and its users, IEEE Trans. Cloud Comput. (2017).
[54] C. Liu, K. Li, J. Liang, K. Li, Cooper-match: Job offloading with a cooperative game for guaranteeing strict deadlines in mec, IEEE Trans. Mob. Comput. (2019).
[55] R.H. Byrd, M.E. Hribar, J. Nocedal, An interior point algorithm for large-scale nonlinear programming, SIAM J. Optim. 9 (4) (1999) 877–900.
[56] R.A. Waltz, J.L. Morales, J. Nocedal, D. Orban, An interior algorithm for nonlinear optimization that combines line search and trust region steps, Math. Program. 107 (3) (2006) 391–408.
[57] P.T. Boggs, J.W. Tolle, Sequential quadratic programming, Acta Numer. 4 (1995) 1–51.
[58] C.T. Lawrence, A.L. Tits, A computationally efficient feasible sequential quadratic programming algorithm, SIAM J. Optim. 11 (4) (2001) 1092–1118.
[59] H.J. Ferreau, C. Kirches, A. Potschka, H.G. Bock, M. Diehl, Qpoases: A parametric active-set algorithm for quadratic programming, Math. Program. Comput. 6 (4) (2014) 327–363.
[60] W.W. Hager, H. Zhang, A new active set algorithm for box constrained optimization, SIAM J. Optim. 17 (2) (2006) 526–557.

Yifan Chen received the M.Sc. degree in the College of Computer Science and Electronic Engineering from Hunan University, Changsha, China, in 2015. She is currently pursuing the Ph.D. degree with the College of Computer Science and Electronic Engineering, Hunan University, China. Her current research interests include Internet of Things, distributed computing, game theory, and mobile edge computing.

Zhiyong Li (M'14) received the M.Sc. degree in system engineering from the National University of Defense Technology, Changsha, China, in 1996, and the Ph.D. degree in control theory and control engineering from Hunan University, Changsha, in 2004. Since 2004, he has been with the College of Computer Science and Electronic Engineering, Hunan University, where he is currently a Full Professor. He has authored over 100 papers in international journals and conferences. His research interests include intelligent perception and autonomous moving body, machine learning and industrial big data, and intelligent optimization algorithms with applications. He is a member of the China Computer Federation and the Chinese Association for Artificial Intelligence.



Bo Yang received the M.Sc. and Ph.D. degree in Computer Science and Technology from Hunan University, Changsha, China, in 2005 and 2018, respectively. Now, he is an assistant Professor with Hunan University of Finance and Economics. His current research interests include parallel computing and high-performance computing, distributed computing, energy-efficient computing and communication, big data computing, game theory, and mobile edge computing. He has served as a reviewer for a number of international journals, including IEEE Transactions on Parallel and Distributed Systems and Journal of Parallel and Distributed Computing.

Ke Nai received the M.Sc. degree in computer science and technology from the College of Information Science and Engineering, Center South University, China, in 2014. He is currently pursuing the Ph.D. degree with the College of Information Science and Engineering, Hunan University, China. His current research interests include visual tracking, face recognition, computer vision, mobile edge computing, and machine learning.

Keqin Li is a SUNY Distinguished Professor of computer science with the State University of New York. He is also a Distinguished Professor at Hunan University, China. His current research interests include cloud computing, fog computing and mobile edge computing, energy-efficient computing and communication, embedded systems and cyber–physical systems, heterogeneous computing systems, big data computing, high-performance computing, CPU–GPU hybrid and cooperative computing, computer architectures and systems, computer networking, machine learning, intelligent and soft computing. He has published over 630 journal articles, book chapters, and refereed conference papers, and has received several best paper awards. He currently serves or has served on the editorial boards of the IEEE Transactions on Parallel and Distributed Systems, the IEEE Transactions on Computers, the IEEE Transactions on Cloud Computing, the IEEE Transactions on Services Computing, and the IEEE Transactions on Sustainable Computing. He is an IEEE Fellow.