A Framework of Price Bidding Configurations for Resource Usage in Cloud Computing

Kenli Li, *Member, IEEE*, Chubo Liu, Keqin Li, *Fellow, IEEE*, and Albert Y. Zomaya, *Fellow, IEEE*

Abstract—In this paper, we focus on price bidding strategies of multiple users competition for resource usage in cloud computing. We consider the problem from a game theoretic perspective and formulate it into a non-cooperative game among the multiple cloud users, in which each cloud user is informed with incomplete information of other users. For each user, we design a utility function which combines the net profit with time efficiency and try to maximize its value. We design a mechanism for the multiple users to evaluate their utilities and decide whether to use the cloud service. Furthermore, we propose a framework for each cloud user to compute an appropriate bidding price. At the beginning, by relaxing the condition that the allocated number of servers can be fractional, we prove the existence of Nash equilibrium solution set for the formulated game. Then, we propose an iterative algorithm ($\mathcal{IA}$), which is designed to compute a Nash equilibrium solution. The convergency of the proposed algorithm is also analyzed and we find that it converges to a Nash equilibrium if several conditions are satisfied. Finally, we revise the obtained solution and propose a near-equilibrium price bidding algorithm ($\mathcal{NPBA}$) to characterize the whole process of our proposed framework. The experimental results show that the obtained near-equilibrium solution is close to the equilibrium one.

Index Terms—Cloud computing, nash equilibrium, non-cooperative game theory, price bidding strategy

1 Introduction

1.1 Motivation

Cloud computing has recently emerged as a new paradigm for a cloud provider to host and deliver computing resources or services to enterprises and consumers [1]. Usually, the provided services mainly refer to Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS), which are all made available to the general public in a pay-as-you-go manner [3], [4]. In most systems, the service provider provides the architecture for multiple users to bid for resource usage [5], [6]. When making bids for resource usage in cloud, multiple users and the cloud provider need to reach an agreement on the service level and the costs to use the provided resources during the reserved time slots, which could lead to a competition for the usage of limited resources [7]. Therefore, it is important for a user to configure an appropriate bidding price for resource usage during his/her reserved time slots without complete information of those other users, such that his/her utility is maximized.

- K. Li and C. Liu are with the College of Information Science and Engineering, Hunan University, and National Supercomputing Center in Changsha, Hunan, China, 410082. E-mail: {lkl, liuchubo}@hnu.edu.cn.
- K. Li is with the College of Information Science and Engineering, Hunan University, and National Supercomputing Center in Changsha, Hunan, China, 410082 and with the Department of Computer Science, State University of New York, New Paltz, New York 12561. E-mail: lik@newpaltz.edu.
- A. Y. Zomaya is with the School of Information Technologies, University of Sydney, Sydney, NSW 2006, Australia. E-mail: albert.zomaya@sydney.edu.au.

Manuscript received 28 July 2015; revised 14 Oct. 2015; accepted 18 Oct. 2015. Date of publication 26 Oct. 2015; date of current version 20 July 2016. Recommended for acceptance by Z. Du.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPDS.2015.2495120

For a cloud provider, the income (i.e., the revenue) is the charge from users for resource usage [8], [9]. When providing computing resources to multiple cloud users, a suitable resource allocation model referring to bidding prices should be significantly taken into account. The reason lies in that an appropriate resource allocation model referring to bidding prices is not just for the profit of a cloud provider, but for the appeals to more cloud users in the market to use cloud service. Specifically, if the per resource usage bidding price is too high, even though the allocated computing resource is enough, a user may refuse to use the cloud service due to the high payment, and choose another cloud provider or just finish his/her requests locally. On the other hand, if the per resource usage charge is low while the allocated computing resource is not sufficiently enough, this will lead to poor service quality (long task response time) and thus dissatisfies the cloud users even for potential users in the market. Hence, a cloud provider should design an appropriate resource allocation model considering users' bidding prices.

A rational user will choose a bidding strategy to use resources that maximizes his/her own net reward, i.e., the utility obtained by choosing the cloud service minus the payment [1]. On the other hand, the utility of a user is not only determined by the importance of his/her tasks (i.e., how much benefit the user can receive by finishing the tasks), but also closely related to the urgency of the task (i.e., how quickly it can be finished). The same task, such as running an online voice recognition algorithm, is able to generate more utility for a cloud user if it can be completed within a shorter period of time in the cloud [1]. However, considering the energy saving and economic reasons, it is irrational for a cloud provider to provide enough computing resources to satisfy all requests in a time slot. Therefore, multiple cloud users have to compete for resource usage. Since the bidding price and allocated computing resources of each user are affected by those decisions of other users, it is natural to analyze the behavior of such systems as a strategic games [10].

1.2 Our Contributions

In this paper, we focus on price bidding strategies of multiple users competition for resource usage in cloud computing. We consider the problem from a game theoretic perspective and formulate it into a non-cooperative game among the multiple cloud users, in which each cloud user is informed with incomplete information of other users. For each user, we design a utility function which combines the net profit with time efficiency and try to maximize its value. We study the conflicts of the multiple users with interactive decisions and propose a near-equilibrium price bidding algorithm ($\mathcal{NPBA}$) to configure appropriate bidding strategy for each of the users. We also perform extensive experiments to verify the effectiveness of our proposed price bidding algorithm. In summary, the main contributions of this work can be listed as follows:

- We propose a framework for each cloud user to configure an appropriate bidding price for resource usage in cloud computing.
- By relaxing the condition that the allocated number of servers can be fractional, we prove the existence of Nash equilibrium solution set for the formulated game and propose an iterative algorithm ($\mathcal{IA}$) to compute a Nash equilibrium solution.
- The convergency of the proposed $\mathcal{IA}$ algorithm is analyzed and we find that it converges to a Nash equilibrium if several conditions are satisfied.
- We revise the obtained solution and propose a near-equilibrium price bidding algorithm to characterize the whole process of our proposed framework.

The experimental results show that the obtained near-equilibrium solution is close to the equilibrium one, which validates the effectiveness of our proposed $\mathcal{NPBA}$ algorithm.

The rest of the paper is organized as follows. In Section 2, we presented the relevant works. Section 3 describes the models of the system and presents the problem to be solved. Section 4 formulates the problem into a non-cooperative game and propose a near-equilibrium price bidding algorithm. Many analyses are also presented in this section. Section 5 is developed to verify our theoretical analysis and show the effectiveness of our proposed algorithm. We conclude the paper with future work in Section 6.

2 Related Works

In many scenarios, a service provider provides the architecture for users to bid for resource usage [6], [11], [12]. One of the most important aspects that should be taken into account by the provider is its resource allocation model referring users bidding prices, which is closely related to its profit and the appeals to market users.

Many works have been done on resource allocation scheme referring to bidding prices in the literature [6], [11], [12], [13], [14], [15]. In [11], Samimi et al. focused on resource allocation in cloud that considers the benefits for both the users and providers. To address the problem, they proposed a new resource allocation model called combinatorial double auction resource allocation (CDARA), which allocates the resources according to bidding prices. In [6], Zaman and Grosu argued that combinatorial auction-based resource allocation mechanisms are especially efficient over the fixed-price mechanisms. They formulated resource allocation problem in clouds as a combinatorial auction problem and proposed two solving mechanisms, which are extensions of two existing combinatorial auction mechanisms. In [12], the authors also presented a resource allocation model using combinatorial auction mechanisms. Similar studies and models can be found in [13], [14], [15], [16]. However, all of these models only try to improve the profits of the cloud providers or cloud users. They failed to configure optimal bidding prices for multiple users or show how their bidding strategies closer to the optimal ones.

Game theory is a field of applied mathematics that describes and analyzes scenarios with interactive decisions [17], [18], [19]. It is a formal study of conflicts and cooperation among multiple competitive users [20] and a powerful tool for the design and control of multiagent systems [21]. There has been growing interest in adopting cooperative and non-cooperative game theoretic approaches to modeling many problems [22], [23], [24], [25]. In [25], Mohsenian-Rad et al. used game theory to solve an energy consumption scheduling problem. In their work, they proved the existence of the unique Nash equilibrium solution and then proposed an algorithm to obtain it. They also analyzed the convergence of their proposed algorithm. Even though the formats for using game theory in our work, i.e., proving Nash equilibrium solution existence, proposing an algorithm, and analyzing the convergence of the proposed algorithm, are similar to [25], the formulated problem and the analysis process are entirely different. In [26], the authors used cooperative and non-cooperative game theory to analyze load balancing for distributed systems. Different from their proposed non-cooperative algorithm, we solve our problem in a distributed iterative way. In our previous work [27], we used non-cooperative game theory to address the scheduling for simple linear deteriorating jobs. For more works on game theory, the reader is referred to [26], [28], [29], [30], [31].

3 System Model and Problem Formulation

To begin with, we present our system model in the context of a service cloud provider with multiple cloud users, and establish some important results. In this paper, we are concerned with a market with a service cloud provider and n cloud users, who are competing for using the computing resources provided by the cloud provider. We denote the set of users as $\mathcal{N} = \{1, \ldots, n\}$. Each cloud user wants to bid for using some servers for several future time slots. The arrival requests from cloud user i ($i \in \mathcal{N}$) is assumed to follow a Poisson process. The cloud provider consists of multiple zones. In each zone, there are many homogeneous servers. In this paper, we focus on the price bidding for resource usage in a same zone and assume that the number of homogeneous servers in the zone is m. The cloud provider tries to allocate cloud user i ($i \in \mathcal{N}$) with m_i servers without violating the constraint $\sum_{i \in \mathcal{N}} m_i \leq m$. The allocated m_i servers for cloud user i ($i \in \mathcal{N}$) are modeled by an M/M/m queue, only serving the requests from user i for t_i future time slots.

We summarize all the notations used in this section in the notation table (see Section 1 of the supplementary material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2015.2495120).

3.1 Bidding Strategy Model

As mentioned above, the n cloud users compete for using the m servers by bidding different strategies. Specifically, each cloud user responds by bidding with a per server usage price p_i (i.e., the payment to use one server in a time slot) and the number of time slots t_i to use cloud service. Hence, the bid of cloud user i $(i \in \mathcal{N})$ is an ordered pair $b_i = \langle p_i, t_i \rangle$.

We assume that cloud user i $(i \in \mathcal{N})$ bids a price $p_i \in \mathcal{P}_i$, where $\mathcal{P}_i = [\underline{p}, \bar{p}_i]$, with $\bar{p}_i$ denoting user i's maximal possible bidding price. $\underline{p}$ is a conservative bidding price, which is determined by the cloud provider. If $\underline{p}$ is greater than $\bar{p}_i$, then $\mathcal{P}_i$ is empty and the cloud user i $(i \in \mathcal{N})$ refuses to use cloud service. As mentioned above, each cloud user i $(i \in \mathcal{N})$ bids for using some servers for t_i future time slots. In our work, we assume that the reserved time slots t_i is a constant once determined by the cloud user i. We define user i's $(i \in \mathcal{N})$ request profile over the t_i future time slots as follow:

$$\boldsymbol{\lambda}_i^{t_i} = \left(\lambda_i^1, \ldots, \lambda_i^{t_i}\right)^T, \tag{1}$$

where λ_i^t $(t \in \mathcal{T}_i)$ with $\mathcal{T}_i = \{1, \ldots, t_i\}$, is the arrival rate of requests from cloud user i in the tth time slot. The arrival of the requests in different time slots of are assumed to follow a Poisson process.

3.2 Server Allocation Model

We consider a server allocation model motivated by [32], [33], where the allocated number of servers is proportional fairness. That is to say, the allocated share of servers is the ratio between the cloud user's product value of his/her bidding price with reserved time slots and the summation of all product values from all cloud users. Then, each cloud user i $(i \in \mathcal{N})$ is allocated a portion of servers as

$$m_i(b_i, \boldsymbol{b}_{-i}) = \left\lfloor \frac{p_i t_i}{\sum_{j \in \mathcal{N}} p_j t_j} \cdot m \right\rfloor, \tag{2}$$

where $\boldsymbol{b}_{-i} = (b_1, \ldots, b_{i-1}, b_{i+1}, \ldots, b_n)$ denotes the vector of all users' bidding profile except that of user i, and $\lfloor x \rfloor$ denotes the greatest integer less than or equal to x. We design a server allocation model as Eq. (2) for two considerations. On one hand, if the reserved time slots to use cloud service t_i is large, the cloud provider can charge less for one server in a unit of time to appeal more cloud users, i.e., the bidding price p_i can be smaller. In addition, for the cloud user i $(i \in \mathcal{N})$, he/she may be allocated more servers, which can improve his/her service time utility. On the other hand, if the bidding price p_i is large, this means that the cloud user i $(i \in \mathcal{N})$ wants to pay more for per server usage in a unit of time to allocate more servers, which can also improve his/her service time utility. This is also beneficial to the cloud provider due to the higher charge for each server. Therefore, we design a server allocation model as Eq. (2), which is proportional to the product of p_i and t_i.

3.3 Cloud Service Model

As mentioned in the beginning, the allocated m_i servers for cloud user i $(i \in \mathcal{N})$ are modeled as an M/M/m queue, only serving the requests from cloud user i for t_i future time slots. The processing capacity of each server for requests from cloud user i $(i \in \mathcal{N})$ is presented by its service rate μ_i. The requests from cloud user i $(i \in \mathcal{N})$ in tth $(t \in \mathcal{T}_i)$ time slot are assumed to follow a Poisson process with average arrival rate λ_i^t.

Let π_{ik}^t be the probability that there are k service requests (waiting or being processed) in the tth time slot and $\rho_i^t = \lambda_i^t/(m_i\mu_i)$ be the corresponding service utilization in the M/M/m queuing system. With reference to [8], we obtain

$$\pi_{ik}^t = \begin{cases} \frac{1}{k!}\left(m_i\rho_i^t\right)^k \pi_{i0}^t, & k < m_i; \\ \frac{m_i^{m_i}\left(\rho_i^t\right)^k}{m_i!}\pi_{i0}^t, & k \geq m_i; \end{cases} \tag{3}$$

where

$$\pi_{i0}^t = \left\{ \sum_{l=0}^{m_i-1} \frac{1}{l!}\left(m_i\rho_i^t\right)^l + \frac{1}{m_i!} \cdot \frac{\left(m_i\rho_i^t\right)^{m_i}}{1-\rho_i^t} \right\}^{-1}. \tag{4}$$

The average number of service requests (in waiting or in execution) in tth time slot is

$$\bar{N}_i^t = \sum_{k=0}^{\infty} k\pi_{ik}^t = \frac{\pi_{im_i}^t}{1-\rho_i^t} = m_i\rho_i^t + \frac{\rho_i^t}{1-\rho_i^t}\Pi_i^t, \tag{5}$$

where Π_i^t represents the probability that the incoming requests from cloud user i $(i \in \mathcal{N})$ need to wait in queue in the tth time slot.

Applying Little's result, we get the average response time in the tth time slot as

$$\bar{T}_i^t = \frac{\bar{N}_i^t}{\lambda_i^t} = \frac{1}{\lambda_i^t}\left(m_i\rho_i^t + \frac{\rho_i^t}{1-\rho_i^t}\Pi_i^t\right). \tag{6}$$

In this work, we assume that the allocated servers for each cloud user will likely keep busy, because if no so, a user can bid lower price to obtain less servers such that the computing resources can be fully utilized. For analytical tractability, Π_i^t is assumed to be 1. Therefore, we have

$$\bar{T}_i^t = \frac{\bar{N}_i^t}{\lambda_i^t} = \frac{1}{\lambda_i^t}\left(m_i\rho_i^t + \frac{\rho_i^t}{1-\rho_i^t}\right) = \frac{1}{\mu_i} + \frac{1}{m_i\mu_i - \lambda_i^t}. \tag{7}$$

Note that the request arrival rate from a user should never exceed the total processing capacity of the allocated servers. In our work, we assume that the remaining processing capacity for serving user i $(i \in \mathcal{N})$ is at least $\sigma\mu_i$, where σ is a relative small positive constant. That is, if $\lambda_i^t > (m_i - \sigma)\mu_i$, cloud user i $(i \in \mathcal{N})$ should reduce his/

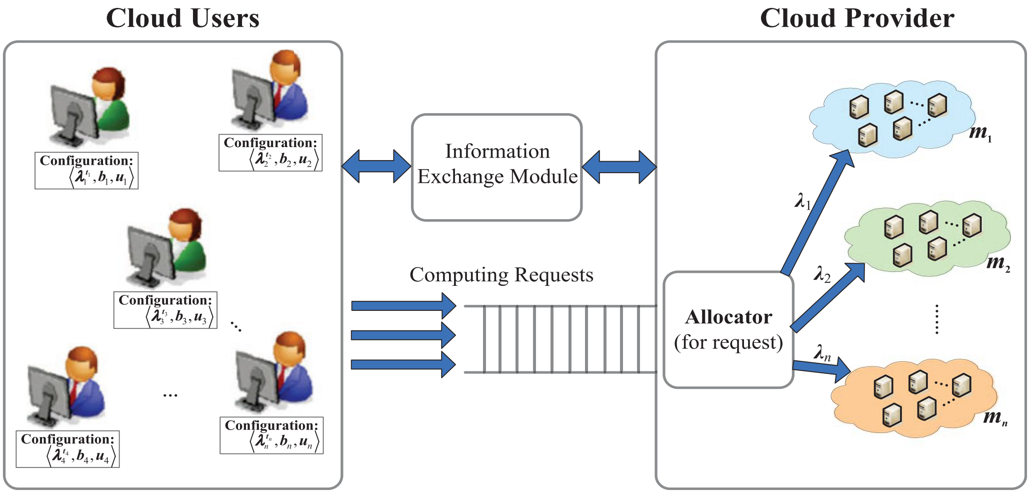


Fig. 1. Architecture model.

her request arrival rate to $(m_i - \sigma)\mu_i$. Otherwise, server crash would be occurred. Hence, we have

$$\bar{T}_i^t = \frac{1}{\mu_i} + \frac{1}{m_i\mu_i - \chi_i^t}, \tag{8}$$

where χ_i^t is the minimum value of λ_i^t and $(m_i - \sigma)\mu_i$, i.e., $\chi_i^t = \min\{\lambda_i^t, (m_i - \sigma)\mu_i\}$.

3.4 Architecture Model

In this section, we model the architecture of our proposed framework to price bids for resource usage in cloud computing. The multiple users can make appropriate bidding decisions through the information exchange module. As shown in Fig. 1, each cloud user i ($i \in \mathcal{N}$) is equipped with a utility function (u_i), the request arrival rate over reserved time slots ($\boldsymbol{\lambda}_i^{t_i}$), and the bidding configuration (b_i), i.e., the payment strategy for one server in a unit of time and the reserved time slots. Let $\Xi_{\mathcal{N}}$ be the aggregated payment from all cloud users for using a server, then we have $\Xi_{\mathcal{N}} = \sum_{i=1}^{n} p_i t_i$. Denote $\boldsymbol{m} = (m_i)_{i\in\mathcal{N}}$ as the server allocation vector, $\boldsymbol{b} = (b_i)_{i\in\mathcal{N}}$ as the corresponding bids, and $\boldsymbol{u} = (u_i)_{i\in\mathcal{N}}$ as the utility functions of all cloud users. The cloud provider consists of m homogeneous servers and communicates some information (e.g., conservative bidding price $\underline{p}$, current aggregated payment from all cloud users for using a server $\Xi_{\mathcal{N}}$) with multiple users through the information exchange module. When multiple users try to make price bidding strategies for resource usage in the cloud provider, they first get information from the information exchange module, then configure proper bidding strategies ($\boldsymbol{b}$) such that their own utilities ($\boldsymbol{u}$) are maximized. After this, they send the updated strategies to the cloud provider. The procedure is terminated when the set of remaining cloud users, who prefer to use the cloud service, and their corresponding bidding strategies are kept fixed.

3.5 Problem Formulation

Now, let us consider user i's ($i \in \mathcal{N}$) utility in time slot t ($t \in \mathcal{T}_i$). A rational cloud user will seek a bidding strategy to maximize his/her expected net reward by finishing the requests, i.e., the benefit obtained by choosing the cloud service minus his/her payment. Since all cloud users are charged based on their bidding prices and allocated number of servers, we denote the cloud user i's payment in time slot t by $P_i^t(b_i, \boldsymbol{b}_{-i})$, where $P_i^t(b_i, \boldsymbol{b}_{-i}) = p_i m_i(b_i, \boldsymbol{b}_{-i})$ with $\boldsymbol{b}_{-i} = (b_1, \ldots, b_{i-1}, b_{i+1}, \ldots, b_n)$ denoting the vector of all users' bidding profile except that of user i. Denote $P_T(b_i, \boldsymbol{b}_{-i})$ as the aggregated payment from all cloud users, i.e., the revenue of the cloud provider. Then, we have

$$P_T(b_i, \boldsymbol{b}_{-i}) = \sum_{i=1}^{n}\sum_{t=1}^{t_i} P_i^t(b_i, \boldsymbol{b}_{-i}) = \sum_{i=1}^{n}(p_i m_i(b_i, \boldsymbol{b}_{-i})t_i). \tag{9}$$

On the other hand, since a user will be more satisfied with much faster service, we also take the average response time into account. From Eq. (8), we know that the average response time of user i ($i \in \mathcal{N}$) is impacted by m_i and χ_i^t, where $\chi_i^t = \min\{\lambda_i^t, (m_i - \sigma)\mu_i\}$. The former is varied by $(b_i, \boldsymbol{b}_{-i})$, and the latter is determined by λ_i^t and m_i. Hence, we denote the average response time of user i as $\bar{T}_i^t(b_i, \boldsymbol{b}_{-i}, \lambda_i^t)$. More formally, the utility of user i ($i \in \mathcal{N}$) in time slot t is defined as

$$u_i^t(b_i, \boldsymbol{b}_{-i}, \lambda_i^t) = r_i\chi_i^t - \delta_i P_i^t(b_i, \boldsymbol{b}_{-i}) - w_i\bar{T}_i^t(b_i, \boldsymbol{b}_{-i}, \lambda_i^t), \tag{10}$$

where χ_i^t is the minimum value of λ_i^t and $(m_i(b_i, \boldsymbol{b}_{-i}) - \sigma)\mu_i$, i.e., $\chi_i^t = \min\{\lambda_i^t, (m_i(b_i, \boldsymbol{b}_{-i}) - \sigma)\mu_i\}$ with σ denoting a relative small positive constant, r_i ($r_i > 0$) is the benefit factor (the reward obtained by finishing one task request) of user i, δ_i ($\delta_i > 0$) is the payment cost factor, and w_i ($w_i > 0$) is the waiting cost factor, which reflects its urgency. If a user i ($i \in \mathcal{N}$) is more concerned with service time utility, then the associated waiting factor w_i might be larger. Otherwise, w_i might be smaller, which implies that the user i is more concerned with profit.

Since the reserved server usage time t_i is a constant and known to cloud user i ($i \in \mathcal{N}$), we use $u_i^t(p_i, \boldsymbol{b}_{-i}, \lambda_i^t)$ instead of $u_i^t(b_i, \boldsymbol{b}_{-i}, \lambda_i^t)$. For further simplicity, we use P_i^t and $\bar{T}_i^t$ to denote $P_i^t(b_i, \boldsymbol{b}_{-i})$ and $T_i^t(b_i, \boldsymbol{b}_{-i}, \lambda_i^t)$, respectively. Following the adopted bidding model, the total utility obtained by user i ($i \in \mathcal{N}$) over all t_i time slots can thus be given by

$$u_i\left(p_i, \boldsymbol{b}_{-i}, \boldsymbol{\lambda}_i^{t_i}\right) = \sum_{t=1}^{t_i} u_i^t(p_i, \boldsymbol{b}_{-i}, \lambda_i^t) = \sum_{t=1}^{t_i}\left(r_i\chi_i^t - P_i^t - w_i\bar{T}_i^t\right). \tag{11}$$

In our work, we assume that each user i ($i \in \mathcal{N}$) has a reservation value v_i. That is to say, cloud user i will prefer to use the cloud service if $u_i(p_i, \boldsymbol{b}_{-i}, \boldsymbol{\lambda}_i^{t_i}) \geq v_i$ and refuse to use the cloud service otherwise.

We consider the scenario where all users are selfish. Specifically, each cloud user tries to maximize his/her total utility over the t_i future time slots, i.e., each cloud user i ($i \in \mathcal{N}$) tries to find a solution to the following optimization problem (OPT$_i$):

$$\text{maximize} \quad u_i\Big(p_i, \boldsymbol{b}_{-i}, \boldsymbol{\lambda}_i^{t_i}\Big), \; p_i \in \mathcal{P}_i. \tag{12}$$

Remark 1. In finding the solution to (OPT$_i$), the bidding strategies of all other users are kept fixed. In addition, the number of reserved time slots once determined by a user is constant. So the variable in (OPT$_i$) is the bidding price of cloud user i, i.e., p_i.

4 Game Formulation and Analyses

In this section, we formulated the considered scenario into a non-cooperative game among the multiple cloud users. By relaxing the condition that the allocated number of servers for each user can be fractional, we analyze the existence of a Nash equilibrium solution set for the formulated game. We also propose an iterative algorithm to compute a Nash equilibrium and then analyze its convergence. Finally, we revise the obtained Nash equilibrium solution and propose an algorithm to characterize the whole process of the framework.

4.1 Game Formulation

Game theory studies the problems in which players try to maximize their utilities or minimize their disutilities. As described in [5], a non-cooperative game consists of a set of players, a set of strategies, and preferences over the set of strategies. In this paper, each cloud user is regarded as a player, i.e., the set of players is the n cloud users. The strategy set of player i ($i \in \mathcal{N}$) is the price bidding set of user i, i.e., $\mathcal{P}_i$. Then the joint strategy set of all players is given by $\mathcal{P} = \mathcal{P}_1 \times \cdots \times \mathcal{P}_n$.

As mentioned before, all users are considered to be selfish and each user i ($i \in \mathcal{N}$) tries to maximize his/her own utility or minimize his/her disutility while ignoring those of the others. Denote

$$\psi_i^t\big(p_i, \boldsymbol{b}_{-i}, \lambda_i^t\big) = \delta_i P_i^t + w_i T_i^t - r_i \chi_{it}. \tag{13}$$

In view of (11), we can observe that user i's optimization problem (OPT$_i$) is equivalent to

$$\begin{aligned} \text{minimize} \quad & f_i\Big(p_i, \boldsymbol{b}_{-i}, \boldsymbol{\lambda}_i^{t_i}\Big) = \sum_{t=1}^{t_i} \psi_i^t\big(p_i, \boldsymbol{b}_{-i}, \lambda_i^t\big), \\ \text{s.t.} \quad & (p_i, \boldsymbol{p}_{-i}) \in \mathcal{P}. \end{aligned} \tag{14}$$

The above formulated game can be formally defined by the tuple $G = \langle \mathcal{P}, \boldsymbol{f} \rangle$, where $\boldsymbol{f} = (f_1, \ldots, f_n)$. The aim of cloud user i ($i \in \mathcal{N}$), given the other players' bidding strategies $\boldsymbol{b}_{-i}$, is to choose a bidding price $p_i \in \mathcal{P}_i$ such that his/her disutility function $f_i\Big(p_i, \boldsymbol{b}_{-i}, \boldsymbol{\lambda}_i^{t_i}\Big)$ is minimized.

Definition 4.1 (Nash equilibrium). *A* Nash equilibrium *of the formulated game $G = \langle \mathcal{P}, \boldsymbol{f} \rangle$ defined above is a price bidding profile $\boldsymbol{p}^*$ such that for every player i ($i \in \mathcal{N}$):*

$$p_i^* \in \underset{p_i \in \mathcal{P}_i}{\arg\min}\, f_i\Big(p_i, \boldsymbol{b}_{-i}, \boldsymbol{\lambda}_i^{t_i}\Big), \; \boldsymbol{p}^* \in \mathcal{P}. \tag{15}$$

At the Nash equilibrium, each player cannot further decrease its disutility by choosing a different price bidding strategy while the strategies of other players are fixed. The equilibrium strategy profile can be found when each player's strategy is the best response to the strategies of other players.

4.2 Nash Equilibrium Existence Analysis

In this section, we analyze the existence of Nash equilibrium for the formulated game $G = \langle \mathcal{P}, \boldsymbol{f} \rangle$ by relaxing one condition that the allocated number of servers for each user can be fractional. Before addressing the equilibrium existence analysis, we show two properties presented in Theorem 4.1 and Theorem 4.2, which are helpful to prove the existence of Nash equilibrium for the formulated game.

Theorem 4.1. *Given a fixed $\boldsymbol{b}_{-i}$ and assuming that $r_i \geq w_i / \big(\sigma^2 \mu_i^2\big)$ ($i \in \mathcal{N}$), then each of the functions $\psi_i^t\big(p_i, \boldsymbol{b}_{-i}, \lambda_i^t\big)$ ($t_i \in \mathcal{T}_i$) is convex in $p_i \in \mathcal{P}_i$.*

Proof. Obviously, $\psi_i^t\big(p_i, \boldsymbol{b}_{-i}, \lambda_i^t\big)$ ($t \in \mathcal{T}_i$) is a real continuous function defined on $\mathcal{P}_i$. The proof of this theorem follows if we can show that $\forall p_{(1)}, p_{(2)} \in \mathcal{P}_i$,

$$\begin{aligned} &\psi_i^t\big(\theta p_{(1)} + (1-\theta) p_{(2)}, \boldsymbol{b}_{-i}, \lambda_i^t\big) \\ &\quad \leq \theta \psi_i^t\big(p_{(1)}, \boldsymbol{b}_{-i}, \lambda_i^t\big) + (1-\theta) \psi_i^t\big(p_{(2)}, \boldsymbol{b}_{-i}, \lambda_i^t\big), \end{aligned}$$

where $0 < \theta < 1$.

Notice that, $\psi_i^t\big(p_i, \boldsymbol{b}_{-i}, \lambda_i^t\big)$ is a piecewise function and the breakpoint satisfies $(m_i - \sigma)\mu_i = \lambda_i^t$. Then, we obtain the breakpoint as

$$p_i^t = \frac{m_i \Xi_{\Xi_{\mathcal{N}\backslash\{i\}}}}{(m - m_i) t_i} = \frac{\big(\lambda_i^t + \sigma \mu_i\big) \Xi_{\mathcal{N}\backslash\{i\}}}{\big((m-\sigma)\mu_i - \lambda_i^t\big) t_i},$$

where $\Xi_{\mathcal{N}\backslash\{i\}}$ denotes the aggregated payment from all cloud users in $\mathcal{N}$ except of user i, i.e., $\Xi_{\mathcal{N}\backslash\{i\}} = \sum_{j \in \mathcal{N}, j \neq i} p_i t_i$. Next, we discuss the convexity of the function $\psi_i^t\big(p_i, \boldsymbol{b}_{-i}, \lambda_i^t\big)$.

Since

$$\psi_i^t\big(p_i, \boldsymbol{b}_{-i}, \lambda_i^t\big) = \delta_i P_i^t + w_i \bar{T}_i^t - r_i \chi_i^t,$$

where $\chi_i^t = \min\big\{(m_i - \sigma)\mu_i, \lambda_i^t\big\}$, we have

$$\frac{\partial \psi_i^t}{\partial p_i}\big(p_i, \boldsymbol{b}_{-i}, \lambda_i^t\big) = \delta_i \frac{\partial P_i^t}{\partial p_i} + w_i \frac{\partial \bar{T}_i^t}{\partial p_i} - r_i \frac{\partial \chi_i^t}{\partial p_i}.$$

On the other hand, since $\frac{\partial \bar{T}_i^t}{\partial p_i} = 0$ for $p_i \in \Big[\underline{p}, p_i^t\Big)$ and $\frac{\partial \chi_i^t}{\partial p_i} = 0$ for $p_i \in \big(p_i^t, \bar{p}_i\big]$, we obtain

$$\frac{\partial}{\partial p_i} \varphi_i^t\big(p_i, \boldsymbol{b}_{-i}, \lambda_i^t\big) = \begin{cases} \delta_i \frac{\partial P_i^t}{\partial p_i} - r_i \frac{\partial \chi_i^t}{\partial p_i}, & p_i < p_i^t; \\ \delta_i \frac{\partial P_i^t}{\partial p_i} + w_i \frac{\partial \bar{T}_i^t}{\partial p_i}, & p_i > p_i^t. \end{cases}$$

Namely,

$$\frac{\partial}{\partial p_i}\varphi_i^t(p_i, \boldsymbol{b}_{-i}, \lambda_i^t) = \begin{cases} \delta_i\left(\frac{mp_it_i\Xi_{\mathcal{N}\backslash\{i\}}}{\Xi_{\mathcal{N}}^2} + m_i\right) - \frac{mr_i\mu_it_i\Xi_{\mathcal{N}\backslash\{i\}}}{\Xi_{\mathcal{N}}^2}, & p_i < p_i^t; \\ \delta_i\left(\frac{mp_it_i\Xi_{\mathcal{N}\backslash\{i\}}}{\Xi_{\mathcal{N}}^2} + m_i\right) - \frac{mw_i\mu_it_i\Xi_{\mathcal{N}\backslash\{i\}}}{(m_i\mu_i-\lambda_i^t)^2\Xi_{\mathcal{N}}^2}, & p_i > p_i^t, \end{cases}$$

where

$$\Xi_{\mathcal{N}} = \Xi_{\mathcal{N}\backslash\{i\}} + p_it_i = \sum\nolimits_{j\in\mathcal{N}} p_jt_j.$$

We can further obtain

$$\frac{\partial^2}{\partial p_i^2}\psi_i^t(p_i, \boldsymbol{b}_{-i}, \lambda_i^t) = \begin{cases} \frac{2mt_i\Xi_{\mathcal{N}\backslash\{i\}}}{\Xi_{\mathcal{N}}^2}\left(\frac{(r_i\mu_i-p_i)t_i}{\Xi_{\mathcal{N}}} + 1\right), & p_i < p_i^t; \\ \frac{2mt_i\Xi_{\mathcal{N}\backslash\{i\}}}{\Xi_{\mathcal{N}}^2}\left(1 - \frac{p_it_i}{\Xi_{\mathcal{N}}}\right) \\ \quad + \frac{2mw_i\mu_it_i^2\Xi_{\mathcal{N}\backslash\{i\}}}{(m_i\mu_i-\lambda_i^t)^2\Xi_{\mathcal{N}}^3}\left(\frac{\mu_i\Xi_{\mathcal{N}\backslash\{i\}}}{(m_i\mu_i-\lambda_i^t)\Xi_{\mathcal{N}}} + 1\right), & p_i > p_i^t. \end{cases}$$

Obviously,

$$\frac{\partial^2}{\partial p_i^2}\psi_i^t(p_i, \boldsymbol{b}_{-i}, \lambda_i^t) > 0,$$

for all $p_i \in [\underline{p}, p_i^t)$ and $p_i \in (p_i^t, \bar{p}_i]$. Therefore, $\forall p_{(1)}, p_{(2)} \in [\underline{p}, p_i^t)$ or $\forall p_{(1)}, p_{(2)} \in (p_i^t, \bar{p}_i]$,

$$\psi_i^t(\theta p_{(1)} + (1-\theta)p_{(2)}, \boldsymbol{b}_{-i}, \lambda_i^t) \le \theta\psi_i^t(p_{(1)}, \boldsymbol{b}_{-i}, \lambda_i^t) + (1-\theta)\psi_i^t(p_{(2)}, \boldsymbol{b}_{-i}, \lambda_i^t),$$

where $0 < \theta < 1$.

Next, we focus on the situation where $p_{(1)} \in [\underline{p}, p_i^t)$ and $p_{(2)} \in (p_i^t, \bar{p}_i]$. Since $\psi_i^t(p_i, \boldsymbol{b}_i, \lambda_i^t)$ is convex on $[\underline{p}, p_i^t)$ and $(p_i^t, \bar{p}_i]$, respectively. We only need to prove that the value of $\psi_i^t(p_i^t, \boldsymbol{b}_i, \lambda_i^t)$ is less than that of in the linear function value connected by the point in $p_{(1)}$ and the point in $p_{(2)}$, i.e.,

$$\psi_i^t(p_i^t, \boldsymbol{b}_i, \lambda_i^t) \le \theta_i^t\psi_i^t(p_{(1)}, \boldsymbol{b}_i, \lambda_i^t) + (1-\theta_i^t)\psi_i^t(p_{(2)}, \boldsymbol{b}_i, \lambda_i^t),$$

where $\theta_i^t = \frac{p_{(2)}-p_i^t}{p_{(2)}-p_{(1)}}$. We proceed as follows (see Fig. (2)).

Define a function $g_i^t(p_i, \boldsymbol{b}_i, \lambda_i^t)$ on $p_i \in \mathcal{P}_i$, where

$$g_i^t(p_i, \boldsymbol{b}_i, \lambda_i^t) = \delta_ip_im_i + \frac{w_i(\sigma+1)}{\sigma\mu_i} - r_i(m_i-\sigma)\mu_i.$$

We have

$$\psi_i^t(p_i, \boldsymbol{b}_i, \lambda_i^t) = g_i^t(p_i, \boldsymbol{b}_i, \lambda_i^t),$$

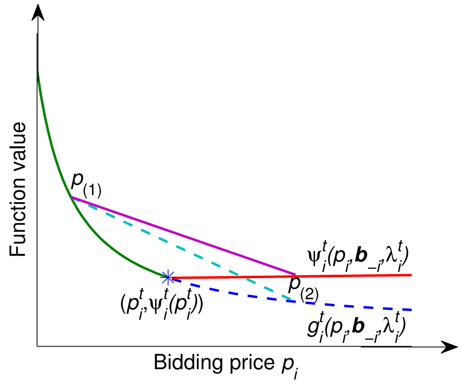


Fig. 2. An illustration.

for all $\underline{p} \le p_i \le p_i^t$. If $r_i \ge w_i/(\sigma^2\mu_i^2)$, then

$$\begin{aligned}\frac{\partial}{\partial p_i}g_i^t(p_i, \boldsymbol{b}_i, \lambda_i^t) &= \delta_i\left(\frac{mp_it_i\Xi_{\mathcal{N}\backslash\{i\}}}{\Xi_{\mathcal{N}}^2} + m_i\right) - \frac{mr_i\mu_it_i\Xi_{\mathcal{N}\backslash\{i\}}}{\Xi_{\mathcal{N}}^2} \\ &\le \delta_i\left(\frac{mp_it_i\Xi_{\mathcal{N}\backslash\{i\}}}{\Xi_{\mathcal{N}}^2} + m_i\right) - \frac{mw_i\mu_it_i\Xi_{\mathcal{N}\backslash\{i\}}}{(m_i\mu_i-\lambda_i^t)^2\Xi_{\mathcal{N}}^2} \\ &= \frac{\partial}{\partial p_i}\psi_i^t(p_i, \boldsymbol{b}_i, \lambda_i^t),\end{aligned}$$

for all $p_i^t < p_i \le \bar{p}_i$. We have

$$\psi_i^t(p_i, \boldsymbol{b}_i, \lambda_i^t) \ge g_i^t(p_i, \boldsymbol{b}_i, \lambda_i^t),$$

for all $p_i^t < p_i \le \bar{p}_i$.

On the other hand, according to the earlier derivation, we know that

$$\frac{\partial^2}{\partial p_i^2}g_i^t(p_i, \boldsymbol{b}_i, \lambda_i^t) > 0,$$

for all $p_i \in \mathcal{P}_i$. That is, $g_i^t(p_i, \boldsymbol{b}_i, \lambda_i^t)$ is a convex function on $\mathcal{P}_i$, and we obtain

$$\begin{aligned}\psi_i^t(p_i^t, \boldsymbol{b}_i, \lambda_i^t) &\le \theta_i^tg_i^t(p_{(1)}, \boldsymbol{b}_i, \lambda_i^t) + (1-\theta_i^t)g_i^t(p_{(2)}, \boldsymbol{b}_i, \lambda_i^t) \\ &= \theta_i^t\psi_i^t(p_{(1)}, \boldsymbol{b}_i, \lambda_i^t) + (1-\theta_i^t)g_i^t(p_{(2)}, \boldsymbol{b}_i, \lambda_i^t) \\ &\le \theta_i^t\psi_i^t(p_{(1)}, \boldsymbol{b}_i, \lambda_i^t) + (1-\theta_i^t)\psi_i^t(p_{(2)}, \boldsymbol{b}_i, \lambda_i^t).\end{aligned}$$

Thus, we have $\psi_i^t(p_i, \boldsymbol{b}_{-i}, \lambda_i^t)$ is convex on $p_i \in \mathcal{P}_i$. This completes the proof and the result follows. □

Theorem 4.2. *If both functions $\mathcal{K}_1(x)$ and $\mathcal{K}_2(x)$ are convex in $x \in \mathcal{X}$, then the function $\mathcal{K}_3(x) = \mathcal{K}_1(x) + \mathcal{K}_2(x)$ is also convex in $x \in \mathcal{X}$.*

Proof. A complete proof of the theorem is given in the supplementary material, available online. □

Theorem 4.3. *There exists a Nash equilibrium solution set for the formulated game $G = \langle\mathcal{P}, \boldsymbol{f}\rangle$, given that the condition* $r_i \ge w_i/(\sigma^2\mu_i^2)$ $(i \in \mathcal{N})$ holds.

Proof. A complete proof of the theorem is given in the supplementary material, available online. □

4.3 Nash Equilibrium Computation

Once we have established that the Nash equilibrium of the formulated game $G = \langle \mathcal{P}, \boldsymbol{f} \rangle$ exists, we are interested in obtaining a suitable algorithm to compute one of these equilibriums with minimum information exchange between the multiple users and the cloud providers.

Note that we can further rewrite the optimization problem (14) as follows:

$$\begin{aligned} \text{minimize} \quad & f_i\Big(p_i, \Xi_{\mathcal{N}}, \boldsymbol{\lambda}_i^{t_i}\Big) = \sum_{t=1}^{t_i} \psi_i^t\big(p_i, \Xi_{\mathcal{N}}, \lambda_i^t\big), \\ \text{s.t.} \quad & (p_i, \boldsymbol{p}_{-i}) \in \mathcal{P}, \end{aligned} \tag{16}$$

where $\Xi_{\mathcal{N}}$ denotes the aggregated payments for each server from all cloud users, i.e., $\Xi_{\mathcal{N}} = \sum_{j \in \mathcal{N}} p_j t_j$. From (16), we can observe that the calculation of the disutility function of each individual user only requires the knowledge of the aggregated payments for a server from all cloud users ($\Xi_{\mathcal{N}}$) rather than that the specific individual bidding strategy profile ($\boldsymbol{b}_{-i}$), which can bring about two advantages. On the one hand, it can reduce communication traffic between users and the cloud provider. On the other hand, it can also keep privacy for each individual user to certain extent, which is seriously considered by many cloud users.

Since all users are considered to be selfish and try to minimize their own disutility while ignoring those of the others. It is natural to consider an iterative algorithm where, at every iteration k, each individual user i ($i \in \mathcal{N}$) updates his/her price bidding strategy to minimize his/her own disutility function $f_i(p_i, \Xi_{\mathcal{N}}, \boldsymbol{\lambda}_i^{t_i})$. The idea is formalized in Algorithm 1.

Algorithm 1. $\mathcal{I}$terative $\mathcal{A}$lgorithm ($\mathcal{IA}$)

Input: $\mathcal{S}$, $\boldsymbol{\lambda}_{\mathcal{S}}$, ϵ.
Output: $\boldsymbol{p}_{\mathcal{S}}$.

//*Initialize* p_i *for each user* $i \in \mathcal{S}$
for (each cloud user $i \in \mathcal{S}$) **do**
  set $p_i^{(0)} \leftarrow \underline{b}$.
end for
Set $k \leftarrow 0$.
//*Find equilibrium bidding prices*
while ($\left\| \boldsymbol{p}_{\mathcal{S}}^{(k)} - \boldsymbol{p}_{\mathcal{S}}^{(k-1)} \right\| > \epsilon$) **do**
  for (each cloud user $i \in \mathcal{S}$) **do**
    Receive $\Xi_{\mathcal{S}}^{(k)}$ from the cloud provider and compute $p_i^{(k+1)}$ as follows (By Algorithm 2):

$$p_i^{(k+1)} \leftarrow \underset{p_i \in \mathcal{P}_i}{\arg\min}\, f_i\Big(p_i, \Xi_{\mathcal{S}}^{(k)}, \boldsymbol{\lambda}_i^{t_i}\Big).$$

    Send the updated price bidding strategy to the cloud provider.
  end for
  Set $k \leftarrow k + 1$.
end while
return $\boldsymbol{p}_{\mathcal{S}}^{(k)}$.

Given $\mathcal{S}$, $\boldsymbol{\lambda}_{\mathcal{S}}$, and ϵ, where $\mathcal{S}$ is the set of cloud users who want to use the cloud service, $\boldsymbol{\lambda}_{\mathcal{S}}$ is the request vector of all cloud users in $\mathcal{S}$, i.e., $\boldsymbol{\lambda}_{\mathcal{S}} = \{\boldsymbol{\lambda}_i^{t_i}\}_{i \in \mathcal{S}}$, and ϵ is a relative small constant. The iterative algorithm finds optimal bidding prices for all cloud users in $\mathcal{S}$. At the beginning of the iterations, the bidding price of each cloud user is set as the conservative bidding price ($\underline{p}$). We use a variable k to index each of the iterations, which is initialized as zero. At the beginning of the iteration k, each of the cloud users i ($i \in \mathcal{N}$) receives the value $\Xi_{\mathcal{S}}^{(k)}$ from the cloud provider and computes his/her optimal bidding price such that his/her own disutility function $f_i(p_i, \Xi_{\mathcal{S}}^{(k)}, \boldsymbol{\lambda}_i^{t_i})$ ($i \in \mathcal{S}$) is minimized. Then, each of the cloud users in $\mathcal{S}$ updates their price bidding strategy and sends the updated value to the cloud provider. The algorithm terminates when the price bidding strategies of all cloud users in $\mathcal{S}$ are kept unchanged, i.e., $\|\boldsymbol{p}_{\mathcal{S}}^{(k)} - \boldsymbol{p}_{\mathcal{S}}^{(k-1)}\| \le \epsilon$.

In subsequent analyses, we show that the above algorithm always converges to a Nash equilibrium if one condition is satisfied for each cloud user. If so, we have an algorithmic tool to compute a Nash equilibrium solution. Before addressing the convergency problem, we first present a property presented in Theorem 4.4, which is helpful to derive the convergence result.

Theorem 4.4. *If* $r_i > \max\{\frac{2\delta_i \bar{p}_i}{\mu_i}, \frac{w_i}{\sigma^2 \mu_i^2}\}$ ($i \in \mathcal{N}$), *then the optimal bidding price* p_i^* ($p_i^* \in \mathcal{P}_i$) *of cloud user* i ($i \in \mathcal{N}$) *is a nondecreasing function with respect to* $\Xi_{\mathcal{N} \setminus \{i\}}$, *where* $\Xi_{\mathcal{N} \setminus \{i\}} = \sum_{j \in \mathcal{N}} p_j t_j - p_i t_i$.

Proof. According to the results in Theorem 4.1, we know that for each cloud user i ($i \in \mathcal{N}$), given a fixed $\boldsymbol{b}_{-i}$, there are t_i breakpoints for the function $f_i(p_i, \boldsymbol{b}_{-i}, \boldsymbol{\lambda}_i^{t_i})$. We denote $\mathcal{B}_i$ as the set of the t_i breakpoints, then we have $\mathcal{B}_i = \{p_i^t\}_{t \in \mathcal{T}_i}$, where

$$p_i^t = \frac{m_i \Xi_{\mathcal{N} \setminus \{i\}}}{(m - m_i) t_i} = \frac{\big(\lambda_i^t + \sigma \mu_i\big) \Xi_{\mathcal{N} \setminus \{i\}}}{\big((m - \sigma)\mu_i - \lambda_i^t\big) t_i}.$$

Combining the above t_i breakpoints with two end points, i.e., $\underline{p}$ and $\bar{p}_i$, we obtain a new set $\mathcal{B}_i \cup \{\underline{p}, \bar{p}_i\}$. Reorder the elements in $\mathcal{B}_i \cup \{\underline{p}, \bar{p}_i\}$ such that $p_i^{(0)} \le p_i^{(1)} \le \cdots \le p_i^{(t_i)} \le p_i^{(t_i+1)}$, where $p_i^{(0)} = \underline{p}$ and $p_i^{(t_i+1)} = \bar{p}_i$. Then, we obtain a new ordered set $\mathcal{B}_i^{'}$. We discuss the claimed theorem by distinguishing three cases according to the first derivative results of the disutility function $f_i(p_i, \boldsymbol{b}_{-i}, \boldsymbol{\lambda}_i^{t_i})$ on $p_i \in \mathcal{P}_i \backslash \mathcal{B}_i$.

Case 1: $\frac{\partial}{\partial \bar{p}_i} f_i(p_i, \boldsymbol{b}_{-i}, \boldsymbol{\lambda}_i^{t_i}) < 0$. According to the results in Theorem 4.2, we know that the second derivative of $f_i(p_i, \boldsymbol{b}_{-i}, \boldsymbol{\lambda}_i^{t_i})$ on $p_i \in \mathcal{P}_i \backslash \mathcal{B}_i$ is positive, i.e., $\frac{\partial^2}{\partial p_i^2} f_i(p_i, \boldsymbol{b}_{-i}, \boldsymbol{\lambda}_i^{t_i}) > 0$ for all $p_i \in \mathcal{P}_i \backslash \mathcal{B}_i$. In addition, if $r_i \ge w_i / \big(\sigma^2 \mu_i^2\big)$, the left partial derivative is less than that of the right partial derivative in each of the breakpoints in $\mathcal{B}_i$. Therefore, if $\frac{\partial}{\partial \bar{p}_i} f_i(p_i, \boldsymbol{b}_{-i}, \boldsymbol{\lambda}_i^{t_i}) < 0$, then $\frac{\partial}{\partial p_i} f_i(p_i, \boldsymbol{b}_{-i}, \boldsymbol{\lambda}_i^{t_i}) < 0$ for all $p_i \in \mathcal{P}_i \backslash \mathcal{B}_i$. Namely, $f_i(p_i, \boldsymbol{b}_{-i}, \boldsymbol{\lambda}_i^{t_i})$ is a decreasing function on $p_i \in \mathcal{P}_i \backslash \mathcal{B}_i$. Hence, the optimal bidding price of cloud user i is

$p_i^* = \bar{p}_i$. That is to say, the bidding price of cloud user i increases with respect to Ξ_{-i}.

Case 2: $\frac{\partial}{\partial \underline{p}} f_i(p_i, \boldsymbol{b}_{-i}, \boldsymbol{\lambda}_i^{t_i}) > 0$. Similar to Case 1, according to the results in Theorem 4.2, we know that $\frac{\partial^2}{\partial p_i^2} f_i(p_i, \boldsymbol{b}_{-i}, \boldsymbol{\lambda}_i^{t_i}) > 0$ for all $p_i \in \mathcal{P}_i \backslash \mathcal{B}_i$. Hence, if $\frac{\partial}{\partial \underline{p}} f_i(p_i, \boldsymbol{b}_{-i}, \boldsymbol{\lambda}_i^{t_i}) > 0$, $f_i(p_i, \boldsymbol{b}_{-i}, \boldsymbol{\lambda}_i^{t_i})$ is an increasing function for all $p_i \in \mathcal{P}_i \backslash \mathcal{B}_i$. Therefore, under this situation, the optimal bidding price of cloud user i is $p_i^* = \underline{p}$, i.e., the optimal bidding price is always the conservative bidding price, which is the initialized value.

Case 3: $\frac{\partial}{\partial \underline{p}} f_i(p_i, \boldsymbol{b}_{-i}, \boldsymbol{\lambda}_i^{t_i}) < 0$ and $\frac{\partial}{\partial \bar{p}_i} f_i(p_i, \boldsymbol{b}_{-i}, \boldsymbol{\lambda}_i^{t_i}) > 0$. Under this situation, it means that there exists an optimal bidding price $p_i^* \in \mathcal{P}_i \backslash \mathcal{B}_i'$ such that

$$\begin{aligned}\frac{\partial}{\partial p_i} f_i\left(p_i^*, \boldsymbol{b}_{-i}, \boldsymbol{\lambda}_i^{t_i}\right) &= \sum_{t=1}^{t_i} \frac{\partial}{\partial p_i} \psi_i^t\left(p_i^*, \boldsymbol{b}_{-i}, \lambda_i^{t_i}\right) \\ &= \sum_{t=1}^{t_i} \left(\frac{\partial P_i^t}{\partial p_i} + w_i \frac{\partial \bar{T}_i^t}{\partial p_i} - r \frac{\partial \chi_i^t}{\partial p_i} \right) = 0. \end{aligned} \tag{17}$$

Otherwise, the optimal bidding price for cloud user i ($i \in \mathcal{N}$) is in $\mathcal{B}_i'$. If above equation holds, then there exists an integer t' ($0 \le t' \le t_i$), such that the optimal bidding price p_i^* is in $(p_i^{(t')}, p_i^{(t'+1)}) \subseteq \mathcal{P}_i \backslash \mathcal{B}_i{}'$.

According to the derivations in Theorem 4.1, we know that the first derivative of $\psi_i^t(p_i, \boldsymbol{b}_{-i}, \lambda_i^t)$ is

$$\frac{\partial}{\partial p_i} \psi_i^t(p_i, \boldsymbol{b}_{-i}, \lambda_i^t) = \begin{cases} \delta_i \left(\frac{m p_i t_i \Xi_{\mathcal{N}\backslash\{i\}}}{\Xi_{\mathcal{N}}^2} + m_i \right) - \frac{m r_i \mu_i t_i \Xi_{\mathcal{N}\backslash\{i\}}}{\Xi_{\mathcal{N}}^2}, & p_i < p_i^t; \\ \delta_i \left(\frac{m p_i t_i \Xi_{\mathcal{N}\backslash\{i\}}}{\Xi_{\mathcal{N}}^2} + m_i \right) - \frac{m w_i \mu_i t_i \Xi_{\mathcal{N}\backslash\{i\}}}{(m_i\mu_i - \lambda_i^t)^2 \Xi_{\mathcal{N}}^2}, & p_i > p_i^t, \end{cases}$$

That is,

$$\frac{\partial}{\partial p_i} \psi_i^t(p_i, \boldsymbol{b}_{-i}, \lambda_i^t) = \begin{cases} \frac{m t_i}{\Xi_{\mathcal{N}}^2} \left(\delta_i p_i (p_i t_i + 2\Xi_{\mathcal{N}\backslash\{i\}}) - r_i u_i \Xi_{\mathcal{N}\backslash\{i\}} \right), & p_i < p_i^t; \\ \frac{m t_i}{\Xi_{\mathcal{N}}^2} \left(\delta_i p_i (p_i t_i + 2\Xi_{\mathcal{N}\backslash\{i\}}) - \frac{w_i u_i \Xi_{\mathcal{N}\backslash\{i\}}}{(m_i\mu_i - \lambda_i^t)^2} \right), & p_i > p_i^t. \end{cases}$$

Therefore, the Eq. (17) is equivalent to the following equation:

$$h(p_i^*) = \sum_{t=1}^{t_i} \varphi_i^t(p_i^*, \boldsymbol{b}_{-i}, \lambda_i^t) = 0,$$

where

$$\varphi_i^t(p_i^*, \boldsymbol{b}_{-i}, \lambda_i^t) = \begin{cases} \delta_i p_i^* (p_i^* t_i + 2\Xi_{\mathcal{N}\backslash\{i\}}) - r_i u_i \Xi_{\mathcal{N}\backslash\{i\}}, & p_i^* < p_i^t; \\ \delta_i p_i^* (p_i^* t_i + 2\Xi_{\mathcal{N}\backslash\{i\}}) - \frac{w_i u_i \Xi_{\mathcal{N}\backslash\{i\}}}{(m_i\mu_i - \lambda_i^t)^2}, & p_i^* > p_i^t. \end{cases}$$

After some algebraic manipulation, we can write the first derivative result of $\varphi_i^t(p_i^*, \boldsymbol{b}_{-i}, \lambda_i^t)$ on p_i^* as

$$\frac{\partial}{\partial p_i^*} \varphi_i^t(p_i^*, \boldsymbol{b}_{-i}, \lambda_i^t) = \begin{cases} 2\delta_i (p_i^* t_i + \Xi_{\mathcal{N}\backslash\{i\}}), & p_i^* < p_i^t; \\ 2\delta_i (p_i^* t_i + \Xi_{\mathcal{N}\backslash\{i\}}) + \frac{2 w_i t_i \mu_i^2 \Xi_{\mathcal{N}\backslash\{i\}}^2}{(m_i\mu_i - \lambda_i^t)^3 \Xi_{\mathcal{N}}^2}, & p_i^* > p_i^t, \end{cases}$$

and the first derivative result of the function $\varphi_i^t(p_i^*, \boldsymbol{b}_{-i}, \lambda_i^t)$ on $\Xi_{\mathcal{N}\backslash\{i\}}$ as

$$\frac{\partial}{\partial \Xi_{\mathcal{N}\backslash\{i\}}} \varphi_i^t(p_i^*, \boldsymbol{b}_{-i}, \lambda_i^t) = \begin{cases} 2\delta_i p_i^* - r_i u_i, & p_i^* < p_i^t; \\ 2\delta_i p_i^* - r_i u_i - \frac{w_i \mu_i}{(m_i\mu_i - \lambda_i^t)^2} - \frac{2 m w_i \mu_i^2 p_i^* t_i \Xi_{\mathcal{N}\backslash\{i\}}}{(m_i\mu_i - \lambda_i^t)^3 \Xi_{\mathcal{N}}^2}, & p_i^* > p_i^t. \end{cases}$$

Obviously, we have

$$\frac{\partial}{\partial p_i^*} \varphi_i^t(p_i^*, \boldsymbol{b}_{-i}, \lambda_i^t) > 0,$$

for all $p_i^* \in \mathcal{P}_i \backslash \mathcal{B}_i'$. If $r_i > 2\delta_i \bar{p}_i / \mu_i$, then

$$\frac{\partial}{\partial \Xi_{\mathcal{N}\backslash\{i\}}} \varphi_i^t(p_i^*, \boldsymbol{b}_{-i}, \lambda_i^t) < 0.$$

Therefore, if $r_i > \max\{\frac{2\delta_i \bar{p}_i}{\mu_i}, \frac{w_i}{\sigma^2 \mu_i^2}\}$, the function $h(b_i^*)$ decreases with the increase of $\Xi_{\mathcal{N}\backslash\{i\}}$. If $\Xi_{\mathcal{N}\backslash\{i\}}$ increases, to maintain the equality $h(b_i^*) = 0$, b_i^* must increase. Hence, b_i^* increases with the increase of $\Xi_{\mathcal{N}\backslash\{i\}}$. This completes the proof and the result follows. □

Theorem 4.5. *Algorithm $\mathcal{IA}$ converges to a Nash equilibrium, given that the condition $r_i > \max\{\frac{2\delta_i \bar{p}_i}{\mu_i}, \frac{w_i}{\sigma^2 \mu_i^2}\}$ ($i \in \mathcal{N}$) holds.*

Proof. We are now ready to show that the proposed $\mathcal{IA}$ algorithm always converges to a Nash equilibrium solution, given that $r_i > \{\frac{2\delta_i \bar{p}_i}{\mu_i}, \frac{w_i}{\sigma^2 \mu_i^2}\}$ ($i \in \mathcal{N}$) holds. Let $p_i^{(k)}$ be the optimal bidding price of cloud user i ($i \in \mathcal{N}$) at the kth iteration. We shall prove above claim by induction that $p_i^{(k)}$ is non-decreasing in k. In addition, since p_i^* is bounded by $\bar{p}_i$, this establishes the result that $p_i^{(k)}$ always converges.

By Algorithm 1, we know that the bidding price of each cloud user is initialized as the conservative bidding price, i.e., $p_i^{(0)}$ is set as $\underline{p}$ for each of the cloud users i ($i \in \mathcal{N}$). Therefore, after the first iteration, we obtain the results $p_i^{(1)} \ge p_i^{(0)}$ for all $i \in \mathcal{N}$. This establishes our induction basis.

Assuming that the result is true in the kth iteration, i.e., $p_i^{(k)} \ge p_i^{(k-1)}$ for all $i \in \mathcal{N}$. Then, we need to show that in the $(k+1)$th iteration, $p_i^{(k+1)} \ge p_i^{(k)}$ is satisfied for all $i \in \mathcal{N}$. We proceed as follows.

By Theorem 4.4, we know that if $r_i > 2\delta_i \bar{p}_i / \mu_i$, the optimal bidding price p_i^* of cloud user i ($i \in \mathcal{N}$) increases with the increase of $\Xi_{\mathcal{N}\backslash\{i\}}$, where $\Xi_{\mathcal{N}\backslash\{i\}} = \sum_{j\in\mathcal{N}, j\neq i} p_j t_j$. In addition, we can deduce that

$$\Xi^{(k)}_{\mathcal{N}\backslash\{i\}} = \sum_{j\in\mathcal{N}, j\neq i} p_j^{(k)} t_j \geq \sum_{j\in\mathcal{N}, j\neq i} p_j^{(k-1)} t_j = \Xi^{(k-1)}_{\mathcal{N}\backslash\{i\}}.$$

Therefore, the optimal bidding price of cloud user i ($i \in \mathcal{N}$) in the $(k+1)$th iteration $p_i^{(k+1)}$, which is a function of $\Xi^{(k)}_{\mathcal{N}\backslash\{i\}}$, satisfies $p_i^{(k+1)} \geq p_i^{(k)}$ for all $i \in \mathcal{N}$. Thus, the result follows. □

Algorithm 2. $\mathcal{C}alculate\_p_i(\Xi, \boldsymbol{\lambda}_i^{t_i}, \epsilon)$

Input: Ξ, $\boldsymbol{\lambda}_i^{t_i}$, ϵ.
Output: p_i^*.

1: Set $t' \leftarrow 0$.
2: //*Find* p_i^* *in* $\mathcal{P}_i\backslash\mathcal{B}_i'$
3: **while** ($t' \leq t_i$) **do**
4: Set $ub \leftarrow p_i^{(t'+1)} - \epsilon$, and $lb \leftarrow p_i^{(t')} + \epsilon$.
5: **if** ($\frac{\partial}{\partial p_i} f_i\left(lb, \Xi, \boldsymbol{\lambda}_i^{t_i}\right) > 0$ or $\frac{\partial}{\partial p_i} f_i\left(ub, \Xi, \boldsymbol{\lambda}_i^{t_i}\right) < 0$) **then**
6: Set $t' \leftarrow t' + 1$; **continue**.
7: **end if**
8: **while** ($ub - lb > \epsilon$) **do**
9: Set $mid \leftarrow (ub + lb)/2$, and $p_i \leftarrow mid$.
10: **if** ($\frac{\partial}{\partial p_i} f_i\left(p_i, \Xi, \boldsymbol{\lambda}_i^{t_i}\right) < 0$) **then**
11: Set $lb \leftarrow mid$.
12: **else**
13: Set $ub \leftarrow mid$.
14: **end if**
15: **end while**
16: Set $p_i \leftarrow (ub + lb)/2$; **break**.
17: **end while**
18: //*Otherwise, find* p_i^* *in* $\mathcal{B}_i'$
19: **if** ($t' = t_i + 1$) **then**
20: Set $\min \leftarrow +\infty$.
21: **for** (each break point $p_i^{(t')} \in \mathcal{B}_i'$) **do**
22: **if** ($f_i\left(p_i^{(t')}, \Xi, \boldsymbol{\lambda}_i^{t_i}\right) < \min$) **then**
23: Set $\min \leftarrow f_i\left(p_i^{(t')}, \Xi, \boldsymbol{\lambda}_i^{t_i}\right)$, and $p_i \leftarrow p_i^{(t')}$.
24: **end if**
25: **end for**
26: **end if**
27: **return** p_i.

Next, we focus on the calculation for the optimal bidding price p_i^* in problem (16), i.e., calculate

$$p_i^* \in \arg\min_{p_i\in\mathcal{P}_i} f_i\left(p_i, \Xi_{\mathcal{N}}, \boldsymbol{\lambda}_i^{t_i}\right). \tag{18}$$

From Theorem 4.5, we know that the optimal bidding price p_i^* of cloud user i ($i \in \mathcal{N}$) is either in $\mathcal{B}_i'$ or in $\mathcal{P}_i\backslash\mathcal{B}_i'$ such that

$$\frac{\partial}{\partial p_i} f_i\left(p_i^*, \Xi_{\mathcal{N}}, \boldsymbol{\lambda}_i^{t_i}\right) = \sum_{t=1}^{t_i} \frac{\partial}{\partial p_i} \psi_i^t\left(p_i^*, \Xi_{\mathcal{N}}, \lambda_i^t\right) = \sum_{t=1}^{t_i} \left(\delta_i \frac{\partial P_i^t}{\partial p_i} + w_i \frac{\partial \bar{T}_i^t}{\partial p_i} - r_i \frac{\partial \chi_i^t}{\partial p_i}\right) = 0, \tag{19}$$

where $\mathcal{B}_i'$ is an ordered set for all elements in $\mathcal{B}_i \cup \{\underline{p}, \bar{p}_i\}$, and $\mathcal{B}_i$ is the set of t_i breakpoints of cloud user i ($i \in \mathcal{N}$), i.e., $\mathcal{B}_i = \left\{p_i^t\right\}_{t\in\mathcal{T}_i}$ with

$$p_i^t = \frac{m_i \Xi_{\mathcal{N}\backslash\{i\}}}{(m - m_i)t_i} = \frac{\left(\lambda_i^t + \sigma\mu_i\right)\Xi_{\mathcal{N}\backslash\{i\}}}{\left((m-\sigma)\mu_i - \lambda_i^t\right)t_i}. \tag{20}$$

Assuming that the elements in $\mathcal{B}_i'$ satisfy $p_i^{(0)} \leq p_i^{(1)} \leq \cdots \leq p_i^{(t_i+1)}$, where $p_i^{(0)} = \underline{p}$ and $p_i^{(t_i+1)} = \bar{p}_i$. If equation (19) holds, then there exists an integer t' ($0 \leq t' \leq t_i$) such that the optimal bidding price $p_i^* \in (p_i^{(t')}, p_i^{(t'+1)}) \subseteq \mathcal{P}_i\backslash\mathcal{B}_i'$. In addition, from the derivations in Theorem 4.5, we know that

$$\frac{\partial^2}{\partial p_i^2} f_i\left(p_i, \Xi_{\mathcal{N}}, \boldsymbol{\lambda}_i^{t_i}\right) > 0, \tag{21}$$

for all $p_i \in \mathcal{P}_i\backslash\mathcal{B}_i'$. Therefore, we can use a binary search method to search the optimal bidding price p_i^* in each of the sets $(p_i^{(t')}, p_i^{(t'+1)}) \subseteq \mathcal{P}_i\backslash\mathcal{B}_i'$ ($0 \leq t_i' \leq t_i$), which satisfies (19). If we cannot find such a bidding price in $\mathcal{P}_i\backslash\mathcal{B}_i'$, then the optimal bidding price p_i^* is in $\mathcal{B}_i'$. The idea is formalized in Algorithm 2.

Given Ξ, $\boldsymbol{\lambda}_i^{t_i}$, and ϵ, where $\Xi = \sum_{j\in\mathcal{N}} p_j t_j$, $\boldsymbol{\lambda}_i^{t_i} = \left\{\lambda_i^t\right\}_{t\in\mathcal{T}_i}$, and ϵ is a relatively small constant. Our optimal price bidding configuration algorithm to find p_i^* is given in Algorithm $\mathcal{C}alculate\_p_i$. The key observation is that the first derivative of function $f_i(p_i, \Xi, \boldsymbol{\lambda}_i^{t_i})$, i.e., $\frac{\partial}{\partial p_i} f_i(p_i, \Xi, \boldsymbol{\lambda}_i^{t_i})$, is an increasing function in $p_i \in (p_i^{(t')}, p_i^{(t'+1)}) \subset \mathcal{P}_i\backslash\mathcal{B}_i'$ (see (21)), where $0 \leq t' \leq t_i$. Therefore, if the optimal bidding price is in $\mathcal{P}_i\backslash\mathcal{B}_i'$, then we can find p_i^* by using the binary search method in one of the intervals $(p_i^{(t')}, p_i^{(t'+1)})$ ($0 \leq t' \leq t_i$) (Steps 3-17). In each of the search intervals $(p_i^{(t')}, p_i^{(t'+1)})$, we set ub as $(p_i^{(t'+1)} - \epsilon)$ and lb as $(p_i^{(t')} + \epsilon)$ (Step 4), where ϵ is relative small positive constant. If the first derivative of function $f_i(p_i, \Xi, \boldsymbol{\lambda}_i^{t_i})$ on lb is positive or the first derivative on ub is negative, then the optimal bidding price is not in this interval (Step 5). Once the interval, which contains the optimal bidding price is decided, we try to find the optimal bidding price p_i^* (Steps 8-16). Notice that, the optimal bidding price may in $\mathcal{B}_i'$ rather than in $\mathcal{P}_i\backslash\mathcal{B}_i'$ (Step 19). Under this situation, we check each of the breakpoints in $\mathcal{B}_i'$ and find the optimal bidding price (Steps 21-25).

By Algorithm 2, we note that the inner while loop (Steps 8-15) is a binary search process, which is very efficient and requires $\Theta(\log \frac{\bar{p}_{\max} - \underline{p}}{\epsilon})$ to complete, where $\bar{p}_{\max}$ is the maximum upper bidding bound of all users, i.e., $\bar{p}_{\max} = \max_{i\in\mathcal{N}}(\bar{p}_i)$. Let $t_{\max} = \max_{i\in\mathcal{N}}(t_i)$, then the outer while loop (Steps 3-17) requires time $\Theta\left(t_{\max} \log \frac{\bar{p}_{\max} - \underline{p}}{\epsilon}\right)$. On the other hand, the for loop (Steps 21-25) requires $\Theta(t_{\max})$ to find solution in set $\mathcal{B}_i'$. Therefore, the time complexity of Algorithm 2 is $\Theta(t_{\max}(\log \frac{\bar{p}_{\max} - \underline{p}}{\epsilon} + 1))$.

4.4 A Near-Equilibrium Price Bidding Algorithm

Notice that, the equilibrium bidding prices obtained by $\mathcal{IA}$ algorithm are considered under the condition that the allocated number servers can be fractional, i.e., in the computation process, we use

$$m_i = \frac{p_i t_i}{\sum_{j \in \mathcal{N}} p_j t_j} \cdot m, \tag{22}$$

instead of

$$m_i = \left\lfloor \frac{p_i t_i}{\sum_{j \in \mathcal{N}} p_j t_j} \cdot m \right\rfloor. \tag{23}$$

Therefore, we have to revise the solution and obtain a near-equilibrium price bidding strategy. Note that, under Eq. (23), there may exist some remaining servers, which is at most n. Considering for this, we reallocate the remaining servers according to the bidding prices. The idea is formalized in our proposed near-equilibrium price bidding algorithm, which characterizes the whole process.

Algorithm 3. $\mathcal{N}$ear-equilibrium $\mathcal{P}$rice $\mathcal{B}$idding $\mathcal{A}$lgorithm ($\mathcal{NPBA}$)

Input: $\mathcal{N}$, $\mathcal{P}$, $\boldsymbol{\lambda}_{\mathcal{N}}$, ϵ.
Output: $\boldsymbol{p}_{\mathcal{N}}$.
1: Set $\mathcal{S}_c \leftarrow \mathcal{N}$, $\mathcal{S}_l \leftarrow \emptyset$, and $k \leftarrow 0$.
2: **while** ($\mathcal{S}_c \neq \mathcal{S}_l$) **do**
3: Set $\boldsymbol{p}_{\mathcal{N}} \leftarrow \boldsymbol{0}$, $\mathcal{S}_l \leftarrow \mathcal{S}_c$, $\boldsymbol{p}_{\mathcal{S}_c} \leftarrow \mathcal{IA}(\mathcal{S}_c, \boldsymbol{\lambda}_{\mathcal{S}_c}, \epsilon)$, and $\Xi \leftarrow \sum_{j \in \mathcal{N}} p_j t_j$.
4: **for** (each cloud user $i \in \mathcal{S}_c$) **do**
5: Compute the allocated servers as (23), i.e., calculate: $m_i \leftarrow \left\lfloor \frac{p_i t_i}{\Xi} \cdot m \right\rfloor$.
6: **end for**
7: Set $m_{\mathcal{R}} \leftarrow m - \sum_{i \in \mathcal{S}_c} m_i$, and $flag \leftarrow$ **true**.
8: **while** ($m_{\mathcal{R}} \neq 0$ and $flag =$ **true**) **do**
9: Set $flag \leftarrow$ **false**.
10: **for** (each cloud user $i \in \mathcal{S}_c$) **do**
11: Compute the reallocated servers, i.e., calculate: $m_i^t \leftarrow \left\lfloor \frac{p_i t_i}{\Xi} \cdot m_{\mathcal{R}} \right\rfloor$.
12: **if** ($u_i(m_i + m_i^t, p_i, \boldsymbol{\lambda}_i^{t_i}) > u_i(m_i, p_i, \boldsymbol{\lambda}_i^{t_i})$) **then**
13: Set $m_i \leftarrow m_i + m_i^t$, $m_{\mathcal{R}} \leftarrow m_{\mathcal{R}} - m_i^t$, and $flag \leftarrow$ false.
14: **end if**
15: **end for**
16: **end while**
17: **for** (each cloud user $i \in \mathcal{S}_c$) **do**
18: **if** ($u_i(m_i, p_i, \boldsymbol{\lambda}_i^{t_i}) < v_i$) **then**
19: Set $p_i \leftarrow 0$, and $\mathcal{S}_c \leftarrow \mathcal{S}_c - \{i\}$.
20: **end if**
21: **end for**
22: **end while**
23: **return** $\boldsymbol{p}_{\mathcal{N}}$.

At the beginning, the cloud provider sets a proper conservative bidding price ($\underline{p}$) and puts its value to into public information exchange module. Each cloud user i ($i \in \mathcal{N}$) sends his/her reserved time slots value (t_i) to the cloud provider. We denote the current set of cloud users who want to use cloud service as $\mathcal{S}_c$ and assume that in the beginning, all cloud users in $\mathcal{N}$ want to use cloud service, i.e., set $\mathcal{S}_c$ as $\mathcal{N}$ (Step 1). For each current user set $\mathcal{S}_c$, we calculate the optimal bidding prices for all users in $\mathcal{S}_c$ by $\mathcal{IA}$ algorithm, under the assumption that the allocated servers can fractional (Step 3). And then, we calculate their corresponding allocated servers (Steps 4-6). We calculate the remaining servers and introduce a $flag$ variable. The inner while loop tries to allocate the remaining servers according to the calculated bidding strategies of the current users in $\mathcal{S}_c$ (Steps 8-16). The variable $flag$ is used to flag whether there is a user in $\mathcal{S}_c$ can improve his/her utility by the allocated number of servers. The while loop terminates until the remaining servers is zero or there is no one such user can improve his/her utility by reallocating the remaining servers. For each user in $\mathcal{S}_c$, if his/her utility value is less than the reserved value, then we assume that he/she refuses to use cloud service (Steps 17-21). The algorithm terminates when the users who want to use cloud service are kept unchanged (Steps 2-22).

TABLE 1
System Parameters

System parameters	(Fixed)–[Varied range] (increment)
Conservative bidding price ($\underline{p}$)	(200)–[200, 540] (20)
Number of cloud users (n)	(100)–[50, 200] (10)
Maximal bidding price ($\bar{p}_i$)	[500, 800]
Market profit factor (r_i)	[30, 120]
Weight value (w_i)	[0.1, 2.5]
Request arrival rates (λ_i^t)	[20, 480]
Processing rate of a server (μ_i)	[60, 120]
Reserving time slots (t_i)	[1, 72]
Reservation value (v_i)	0
Payment cost weight (δ_i)	1
Other parameters (ϵ, σ, m)	(0.01, 0.1, 600)

5 Performance Evaluation

In this section, we provide some numerical results to validate our theoretical analyses and illustrate the performance of the $\mathcal{NPBA}$ algorithm.

In the following simulation results, we consider the scenario consisting of maximal 200 cloud users. Each time slot is set as one hour of a day and the maximal time slots of a user can be 72. As shown in Table 1, the conservative bidding price ($\underline{p}$) is varied from 200 to 540 with increment 20. The number of cloud users (n) is varied from 50 to 200 with increment 10. The maximal bidding price ($\bar{p}_i$) and market benefit factor (r_i) of each cloud user are randomly chosen from 500 to 800 and 30 to 120, respectively. Each cloud user i ($i \in \mathcal{N}$) chooses a weight value from 0.1 to 2.5 to balance his/her time utility and profit. We assume that the request arrival rate (λ_i^t) in each time slot of each cloud user is selected randomly and uniformly between 20 and 480. The processing rate (μ_i) of a server to the requests from cloud user i ($i \in \mathcal{N}$) is randomly chosen from 60 to 120. For simplicity, the reservation value (v_i) and payment cost weight (δ_i) for each of the cloud users are set as zero and one, respectively. The number of servers m in the cloud provider is set as a constant 600, σ is set as 0.1, and ϵ is set as 0.01.

Fig. 3 shows an instance for the bidding prices of six different cloud users versus the number of iterations of the proposed $\mathcal{IA}$ algorithm. Specifically, Fig. 3 presents the bidding price results of six randomly selected cloud users (users 8, 18, 27, 41, 59, and 96) with a scenario consisting of 100 cloud users. We can observe that the bidding prices of

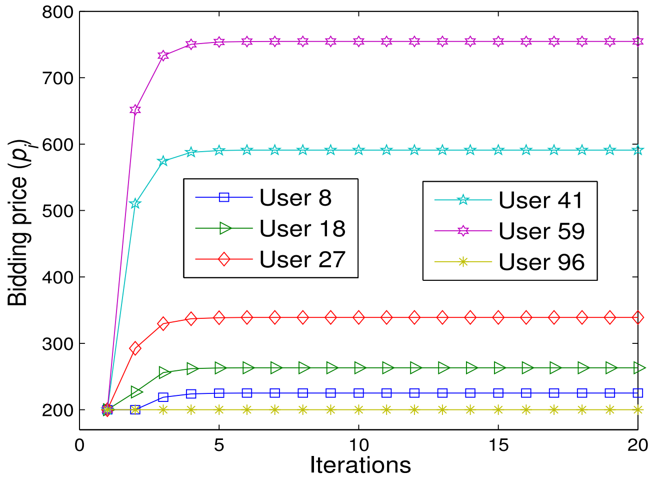


Fig. 3. Convergence process of bidding price.

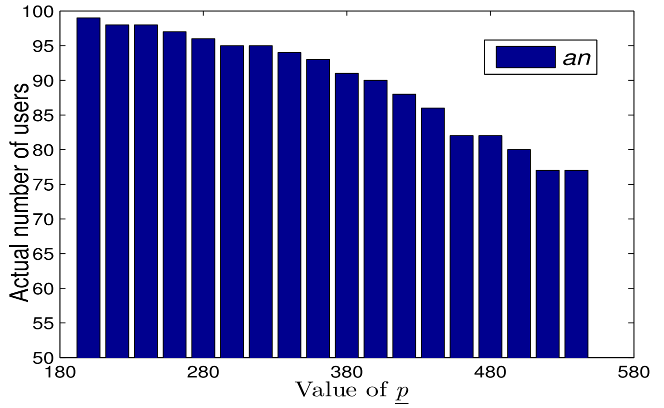


Fig. 5. Actual number of cloud users.

all users seem to be non-decreasing with the increase of iteration number and finally reach a relative stable state, which verifies the validness of Theorem 3.4. That is, the bidding prices of all cloud users keep unchanged, i.e., reach a Nash equilibrium solution after several iterations. In addition, it can also be seen that the developed algorithm converges to a Nash equilibrium very quickly. Specifically, the bidding price of each user has already achieved a relatively stable state after five iteration, which shows the high efficiency of our developed algorithm.

In Fig. 4, we show the trend of the aggregated payment from all cloud users (P_T), i.e., the revenue of the cloud provider, versus the increment of the conservative bidding price. We compare two kinds of results with the situations by computing the allocated number of servers for each cloud user i ($i \in \mathcal{N}$) as (22) and (23), respectively. Specifically, we denote the obtained payment as V_T when compute m_i as (22) and P_T for (23). Obviously, the former is the optimal value computed from the Nash equilibrium solution and bigger than that of the latter. However, it cannot be applied in a real application, because the allocated number of servers cannot be fractional. We just obtain a near-equilibrium solution by assuming that the allocated number of servers can be fractional at first. Even though the obtained solution is not optimal, we can compare these two kinds of results and show that how closer our proposed algorithm can find a near-equilibrium solution to that of the computed optimal one.

We can observe that the aggregated payment from all cloud users tends to increase with the increase of conservative bidding price at first. However, it decreases when conservative bidding price exceeds a certain value. The reason behind lies in that when conservative bidding price increases, more and more cloud users refuse to use the cloud service due to the conservative bidding price exceeds their possible maximal price bidding values or their utilities are less than their reservation values, i.e., the number of users who choose cloud service decreases (see Fig. 5). We can also observe that the differences between the values of P_T and V_T are relatively small and make little differences with the increase of the conservative bidding price. Specifically, the percent differences between the values of V_T and P_T range from 3.99 to 8.41 percent, which reflects that our $\mathcal{NPBA}$ algorithm can find a very well near-optimal solution while ignoring the increment of conservative bidding price. To demonstrate this phenomenon, we further investigate the specific utilities of some users and their corresponding bidding prices, which are presented in Figs. 6 and 7.

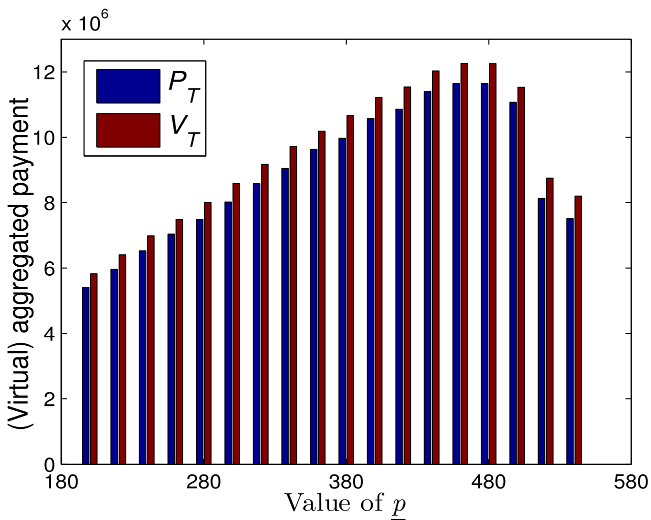


Fig. 4. Aggregated payment of all users.

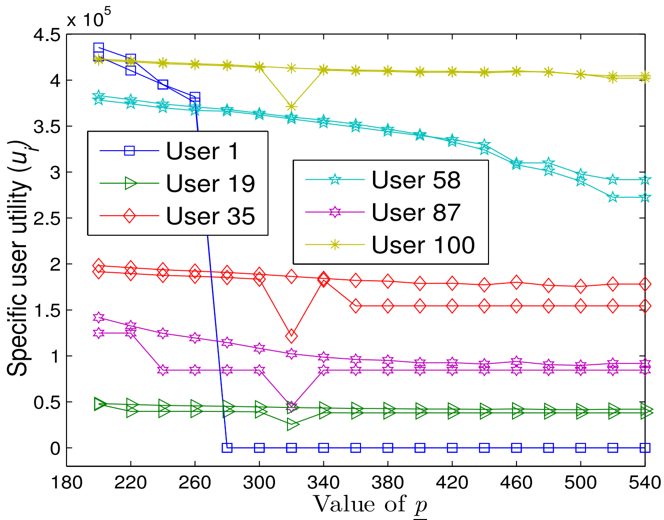


Fig. 6. Specific user utility.

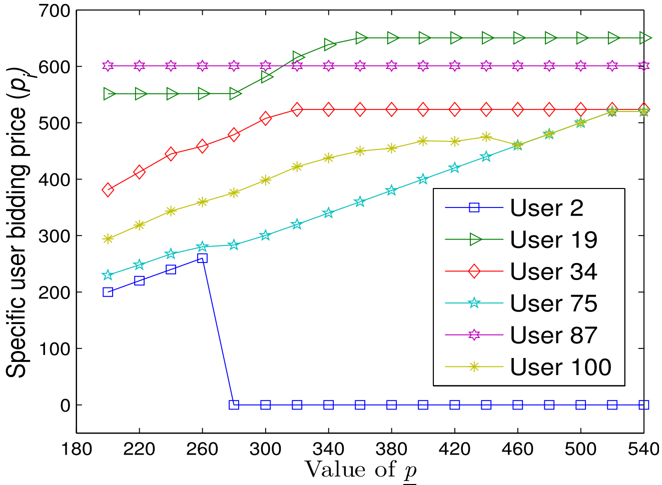


Fig. 7. Specific user bidding price.

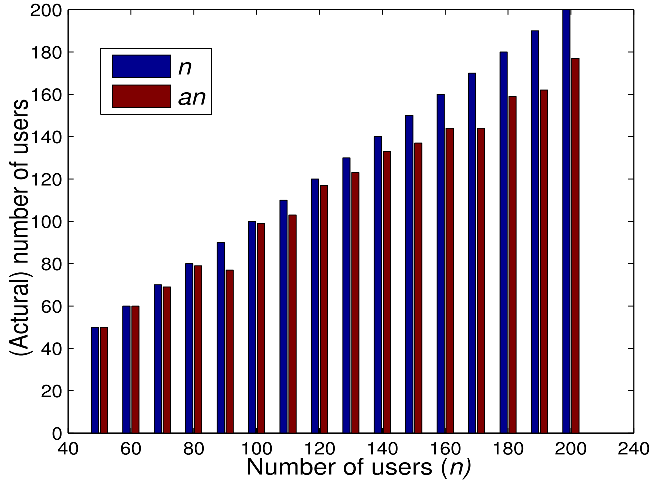


Fig. 9. (Actural) number of cloud users.

In Figs. 6 and 7, we plot the utility shape and the bidding prices of some cloud users for the developed $\mathcal{NPBA}$ algorithm. Fig. 6 presents the utility shape under the developed algorithm versus the increment of conservative bidding price. We randomly select six users (users 1, 19, 35, 58, 87, and 100). It can be seen that the utility trends of all cloud users tend to decreases with the increase of conservative bidding price. However, under every conservative bidding price, for each user, the differences between the utilities computed by using m_i as (22) (the larger one) and (23) (the smaller one) for each cloud user are relatively small. Therefore, the differences between the aggregated payments of (P_T) and (V_T) are small (see Fig. 4). Fig. 7 exhibits the corresponding bidding prices of the users shown in Fig. 6. We can observe that some users may refuse to use cloud service when conservative bidding price exceeds a certain value (user 2). When users choose to use cloud service, the treads of their bidding prices tend to be non-decreasing with the increment of conservative bidding price (user 19, 34, 75, 87, and 100). This phenomenon also verifies the aggregated payment trend shown in Fig. 4. Specifically, due to the increases of users' bidding prices, the aggregated payment from all cloud users tend to increase at first. However, when conservative bidding price exceeds a certain value, more and more cloud users refuse to use cloud service. Therefore, the aggregated payment tends to decrease when conservative bidding price is large enough.

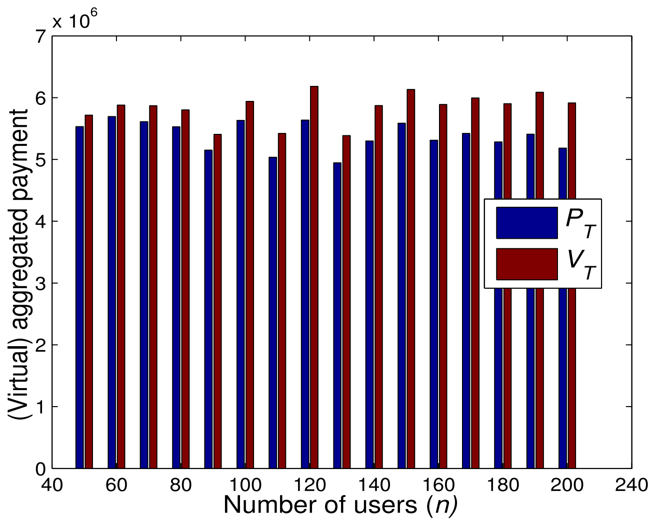


Fig. 8. Aggregated payment on number of users.

In Fig. 8, we show the impact of number of cloud users on aggregated payment. Similar to Fig. 4, the differences between the values of P_T and V_T are relatively small. Specifically, the percent differences between the values of V_T and P_T range from 3.14 to 12.37 percent. That is, the aggregated payment results for different number of users are largely unchanged. In Fig. 9, we can observe that with the increase of number of cloud users, the trend of the differences between the number of cloud users and the actual number of cloud users who choose cloud service also increases. The reason behind lies in that with the increase of number of cloud users, more and more users refuse to use cloud service due to their utilities are less than their conservative values. This also partly verifies the aggregated payment trend shown in Fig. 8, in which the aggregated payments are largely unchanged with the increase of number cloud users.

6 Conclusions

With the popularization of cloud computing and its many advantages such as cost-effectiveness, flexibility, and scalability, more and more applications are moved from local to cloud. However, most cloud providers do not provide a mechanism in which the users can configure bidding prices and decide whether to use the cloud service. To remedy these deficiencies, we focus on proposing a framework to obtain an appropriate bidding price for each cloud user.

We consider the problem from a game theoretic perspective and formulate it into a non-cooperative game among the multiple cloud users, in which each cloud user is informed with incomplete information of other users. For each user, we design a utility function which combines the net profit with time efficiency and try to maximize its value. We design a mechanism for the multiple users to evaluate their utilities and decide whether to use the cloud service. Furthermore, we propose a framework for each cloud user to compute an appropriate bidding price. At the beginning, by relaxing the condition that the allocated number of servers can be fractional, we prove the existence of Nash equilibrium solution set for the formulated game. Then, we propose an iterative algorithm, which is designed to compute a Nash equilibrium solution. The convergency of the proposed algorithm is also analyzed and we find that it converges to a Nash equilibrium if several conditions are satisfied. Finally, we revise the obtained solution and propose a near-equilibrium price bidding algorithm to characterize the whole process of our proposed framework. The experimental results show that the obtained near-equilibrium solution is close to the equilibrium one.

As part of future directions, we will configure the multiple servers in cloud dynamically and study the relationship between the cloud provider and multiple users. Another direction is to study the cloud choice among multiple different cloud providers or determine a proper mixed bidding strategy.

Acknowledgments

The research was partially funded by the Key Program of National Natural Science Foundation of China (Grant Nos. 61133005, 61432005), the National Natural Science Foundation of China (Grant Nos. 61370095, 61472124, 61402400), and the International Science & Technology Cooperation Program of China (Grant No. 2015DFA11240). Chubo Liu is the corresponding author.

References

[1] Y. Feng B. Li, and B. Li, "Price competition in an oligopoly market with multiple iaas cloud providers," *IEEE Trans. Comput.*, vol. 63, no. 1, pp. 59–73, Jan. 2014.

[2] H. Chen, Y. Li, R. H. Louie, and B. Vucetic, "Autonomous demand side management based on energy consumption scheduling and instantaneous load billing: An aggregative game approach," *IEEE Trans. Smart Grid*, vol. 5, no. 4, pp. 1744–1754, Jul. 2014.

[3] R. Pal and P. Hui, "Economic models for cloud service markets: Pricing and capacity planning," *Theoretical Comput. Sci.*, vol. 496, pp. 113–124, 2013.

[4] P. D. Kaur and I. Chana, "A resource elasticity framework for qos-aware execution of cloud applications," *Future Gener. Comput. Syst.*, vol. 37, pp. 14–25, 2014.

[5] S. Tang, J. Yuan, C. Wang, and X.-Y. Li, "A framework for amazon ec2 bidding strategy under sla constraints," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 1, pp. 2–11, Jan. 2014.

[6] S. Zaman and D. Grosu, "Combinatorial auction-based allocation of virtual machine instances in clouds," *J. Parallel Distrib. Comput.*, vol. 73, no. 4, pp. 495–508, Apr. 2013.

[7] S. Son and K. M. Sim, "A price- and-time-slot-negotiation mechanism for cloud service reservations," *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.*, vol. 42, no. 3, pp. 713–728, Jun. 2012.

[8] J. Cao, K. Hwang, K. Li, and A. Y. Zomaya, "Optimal multiserver configuration for profit maximization in cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 6, pp. 1087–1096, Jun. 2013.

[9] J. Mei, K. Li, A. Ouyang, and K. Li, "A profit maximization scheme with guaranteed quality of service in cloud computing," *IEEE Trans. Comput.*, vol. 64, no. 11, pp. 3064–3078, 2015.

[10] E. Krpeolu, A. en, and K. Gler, "Non-cooperative joint replenishment under asymmetric information," *Eur. J. Oper. Res.*, vol. 227, no. 3, pp. 434–443, 2013.

[11] P. Samimi, Y. Teimouri, and M. Mukhtar, "A combinatorial double auction resource allocation model in cloud computing," *Inf. Sci.*, 2014.

[12] T. T. Huu and C.-K. Tham, "An auction-based resource allocation model for green cloud computing," in *Proc. IEEE Int. Conf. Cloud Eng.*, Mar. 2013, pp. 269–278.

[13] M. Rahman and R. Rahman, "Capmauction: Reputation indexed auction model for resource allocation in grid computing," in *Proc. 7th Int. Conf. Elect. Comput. Eng.*, Dec. 2012, pp. 651–654.

[14] A. Ozer and C. Ozturan, "An auction based mathematical model and heuristics for resource co-allocation problem in grids and clouds," in *Proc. 5th Int. Conf. Soft Comput., Comput. Words Perceptions Syst. Anal., Decision Control.*, Sep. 2009, pp. 1–4.

[15] X. Wang, X. Wang, C.-L. Wang, K. Li, and M. Huang, "Resource allocation in cloud environment: A model based on double multi-attribute auction mechanism," in *Proc. IEEE 6th Int. Conf. Cloud Comput. Technol. Sci.*, Dec. 2014, pp. 599–604.

[16] X. Wang, X. Wang, H. Che, K. Li, M. Huang, and C. Gao, "An intelligent economic approach for dynamic resource allocation in cloud services," *IEEE Trans. Cloud Comput.*, vol. 3, no. 3, pp. 275–289, 2015.

[17] G. Scutari, D. Palomar, F. Facchinei, and J.-S. Pang, "Convex optimization, game theory, and variational inequality theory," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 35–49, May 2010.

[18] M. J. Osborne and A. Rubinstein, *A Course in Game Theory*. Cambridge, MA, USA: MIT Press, 1994.

[19] J.-P. Aubin, *Mathematical Methods of Game and Economic Theory*. New York, NY, USA: Dover, 2007.

[20] S. S. Aote and M. Kharat, "A game-theoretic model for dynamic load balancing in distributed systems," in *Proc. Int. Conf. Adv. Comput. Commun. Control*, 2009, pp. 235–238.

[21] N. Li and J. Marden, "Designing games for distributed optimization," in *Proc. 50th IEEE Conf. Decision Control Eur. Control Conf.*, Dec. 2011, pp. 2434–2440.

[22] E. Tsiropoulou, G. Katsinis, and S. Papavassiliou, "Distributed uplink power control in multiservice wireless networks via a game theoretic approach with convex pricing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 1, pp. 61–68, Jan. 2012.

[23] G. Scutari and J.-S. Pang, "Joint sensing and power allocation in nonconvex cognitive radio games: Nash equilibria and distributed algorithms," *IEEE Trans. Inf. Theory*, vol. 59, no. 7, pp. 4626–4661, Jul. 2013.

[24] N. Immorlica, L. E. Li, V. S. Mirrokni, and A. S. Schulz, "Coordination mechanisms for selfish scheduling," *Theoretical Comput. Sci.*, vol. 410, no. 17, pp. 1589–1598, 2009.

[25] A.-H. Mohsenian-Rad, V. W. Wong, J. Jatskevich, R. Schober, and A. Leon-Garcia, "Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid," *IEEE Trans. Smart Grid*, vol. 1, no. 3, pp. 320–331, Dec. 2010.

[26] S. Penmatsa and A. T. Chronopoulos, "Game-theoretic static load balancing for distributed systems," *J. Parallel Distrib. Comput.*, vol. 71, no. 4, pp. 537–555, 2011.

[27] K. Li, C. Liu, and K. Li, "An approximation algorithm based on game theory for scheduling simple linear deteriorating jobs," *Theoretical Comput. Sci.*, vol. 543, no. 0, pp. 46–51, 2014.

[28] I. Atzeni, L. G. Ordóñez, G. Scutari, D. P. Palomar, and J. R. Fonollosa, "Noncooperative and cooperative optimization of distributed energy generation and storage in the demand-side of the smart grid," *IEEE Trans. Signal Process.*, vol. 61, no. 10, pp. 2454–2472, May 2013.

[29] N. Mandayam, G. Editor, S. Wicker, J. Walrand, T. Basar, J. Huang, and D. Palomar, "Game theory in communication systems [guest editorial]," *IEEE Trans. Sel. Areas Commun.*, vol. 26, no. 7, pp. 1042–1046, Sep. 2008.

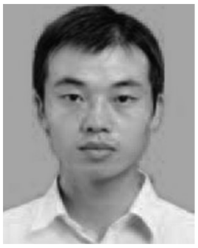
[30] E. Larsson, E. Jorswieck, J. Lindblom, and R. Mochaourab, "Game theory and the flat-fading Gaussian interference channel," *IEEE Signal Process. Mag.*, vol. 26, no. 5, pp. 18–27, Sep. 2009.

[31] C. Liu, K. Li, C. Xu, and K. Li, "Strategy configurations of multiple users competition for cloud service reservation," *IEEE Trans. Parallel Distrib. Syst.*, Doi: 10.1109/TPDS.2015.2398435.

[32] K. Akkarajitsakul, E. Hossain, and D. Niyato, "Distributed resource allocation in wireless networks under uncertainty and application of Bayesian game," *IEEE Commun. Mag.*, vol. 49, no. 8, pp. 120–127, Aug. 2011.

[33] S. Misra, S. Das, M. Khatua, and M. Obaidat, "Qos-guaranteed bandwidth shifting and redistribution in mobile cloud environment," *IEEE Trans. Cloud Comput.*, vol. 2, no. 2, pp. 181–193, Apr. 2014.

Kenli Li received the PhD degree in computer science from the Huazhong University of Science and Technology, China, in 2003. He was a visiting scholar at the University of Illinois at Urbana-Champaign from 2004 to 2005. He is currently a full professor of computer science and technology at Hunan University and deputy director of the National Supercomputing Center in Changsha. His major research areas include parallel computing, high-performance computing, grid and cloud computing. He has published more than 130 research papers in international conferences and journals such as *IEEE Transactions on Computers*, *IEEE Transactions on Parallel and Distributed Systems*, *IEEE Transactions on Signal Processing*, *Journal of Parallel and Distributed Computing*, ICPP, CCGrid. He is an outstanding member of CCF. He is a member of the IEEE and serves on the editorial board of *IEEE Transactions on Computers*.



Chubo Liu is currently working toward the PhD degree at Hunan University, China. His research interests are mainly in modeling and scheduling of distributed computing systems, approximation and randomized algorithms, game theory, grid and cloud computing.

Keqin Li is a SUNY distinguished professor of computer science. His current research interests include parallel computing and high-performance computing, distributed computing, energy-efficient computing and communication, heterogeneous computing systems, cloud computing, big data computing, CPU-GPU hybrid and cooperative computing, multicore computing, storage and file systems, wireless communication networks, sensor networks, peer-to-peer file sharing systems, mobile computing, service computing, Internet of things, and cyber-physical systems. He has published more than 370 journal articles, book chapters, and refereed conference papers, and has received several best paper awards. He is currently or has served on the editorial boards of *IEEE Transactions on Parallel and Distributed Systems*, *IEEE Transactions on Computers*, *IEEE Transactions on Cloud Computing*, *and Journal of Parallel and Distributed Computing*. He is a fellow of the IEEE.

Albert Y. Zomaya is currently the chair professor of high-performance computing and networking and Australian research council professorial fellow in the School of Information Technologies, University of Sydney. He is also the director of the Centre for Distributed and High Performance Computing which was established in late 2009. He is the author/co-author of seven books, more than 400 papers, and the editor of nine books and 11 conference proceedings. He was the editor-in-chief of the *IEEE Transactions on Computers* and serves as an associate editor for 19 leading journals, such as, the *IEEE Transactions on Parallel and Distributed Systems* and *Journal of Parallel and Distributed Computing*. He received the Meritorious Service Award, in 2000 and the Golden Core Recognition, in 2006, both from the IEEE Computer Society. Also, he received the IEEE Technical Committee on Parallel Processing Outstanding Service Award and the IEEE Technical Committee on Scalable Computing Medal for Excellence in Scalable Computing, both in 2011. He is a chartered engineer, a fellow of AAAS, IEEE, and IET United Kingdom.