

# The Value of Cooperation: Minimizing User Costs in Multi-Broker Mobile Cloud Computing Networks

Zhangyu Guan, *Member, IEEE*, and Tommaso Melodia, *Member, IEEE*

**Abstract**—We study the problem of user cost minimization in mobile cloud computing (MCC) networks. We consider a MCC model where multiple brokers assign cloud resources to mobile users. The model is characterized by an heterogeneous cloud architecture (which includes a public cloud and a cloudlet) and by the heterogeneous pricing strategies of cloud service providers. In this setting, we investigate two classes of cloud reservation strategies, i.e., a competitive strategy, and a compete-then-cooperate strategy as a performance bound. We first study a purely competitive scenario where brokers compete to reserve computing resources from remote public clouds (which are affected by long delays) and from local cloudlets (which have limited computational resources but short delays). We provide theoretical results demonstrating the existence of disagreement points (i.e., the equilibrium reservation strategy that no broker has incentive to deviate unilaterally from) and convergence of the best-response strategies of the brokers to disagreement points. We then consider the scenario in which brokers agree to cooperate in exchange for a lower average cost of resources. We formulate a cooperative problem where the objective is to minimize the total average price of all brokers, under the constraint that no broker should pay a price higher than the disagreement price (i.e., the competitive price). We design new globally optimal solution algorithm to solve the resulting non-convex cooperative problem, based on a combination of the *branch and bound framework* and of *advanced convex relaxation techniques*. The resulting optimal solution provides a lower bound on the achievable user cost *without complete collusion* among brokers. Compared with pure competition, we found that (i) noticeable cooperative gains can be achieved over pure competition in markets with a few brokers only, and (ii) the cooperative gain is only marginal in crowded markets, i.e., with a high number of brokers, hence there is no clear incentive for brokers to cooperate.

**Index Terms**—Mobile cloud computing, heterogeneous cloud architecture, resource allocation, game theory

## 1 INTRODUCTION

MOBILE cloud computing (MCC) is emerging as a technology with a potential to provide high-quality and multimedia-rich services in mobile environments [1], [2]. Example applications include cloud-assisted video encoding, multimedia rendering for interactive gaming, mobile healthcare, electronic commerce, and mobile learning [3]. Through MCC technology, mobile devices can continuously offload and run computationally-intensive tasks on “cloud” servers. For example, “smart” phones, cameras, or glasses in need to compress a captured video sequence may perform complex motion-estimation or optimal rate allocation in the cloud. This may enhance the capabilities of mobile devices, hence potentially extending their battery lifetime. However, accessing cloud computational resources in future multimedia-rich MCC networks may result in significant additional costs to users [4]. Therefore, in this paper we focus on the problem of cost minimization for mobile users in future MCC networks.

*Heterogeneity of cloud architectures.* Two main challenges have to be addressed, which are brought about by (i) the *heterogeneity of the cloud architecture* and (ii) the *heterogeneity of the pricing strategies of different cloud providers*.

Future MCC networks will be based on highly heterogeneous architectures. As of today, multiple different cloud architectures have been deployed or proposed to support an emerging wide range of innovative cloud services. These include so-called *public clouds*, *cloudlets*, and their combinations. Public clouds, including Amazon EC2, Microsoft Azure, and Apple iCloud are usually deployed and maintained by large organizations. While they provide computing services with high reliability, scalability, and elasticity, their geographical deployment is typically sparse. For example, Amazon EC2 deployed its data centers only in eight major regions across the world (including US East, US West, Asia Pacific, among others) [5]. Hence, the resulting user-to-cloud delay can be potentially high, which is certainly undesirable for time-sensitive applications. The concept of *cloudlet* has been then introduced to mitigate this problem. Different from public clouds, cloudlets are typically resource-rich computers or clusters, and are operated by much smaller organizations, like a campus or a coffee shop, to provide computing services to only a few users at a time [6]. Since cloudlets are usually deployed at the edge of a wide area network (WAN), and hence close to the end users, the user-to-cloudlet delay is typically negligible compared to the public clouds. We envision that public clouds

- Z. Guan is with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115. E-mail: zguan@ece.neu.edu.
- T. Melodia is with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115. E-mail: melodia@ece.neu.edu.

Manuscript received 24 Mar. 2014; revised 15 Jan. 2015; accepted 24 Apr. 2015. Date of publication 2 June 2015; date of current version 6 Dec. 2017.

Recommended for acceptance by H. Xu.

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/TCC.2015.2440257

and cloudlets will coexist in emerging MCC networks. To meet the increasing quality of experience (QoE) requirements of end users, it is therefore necessary to study tradeoffs between accessing the public cloud with high scalability but potential longer delays, and accessing the cloudlet with lower delay but limited availability of computing resources.

*Heterogeneity of pricing schemes.* Second, users may be charged in several different ways to access computing resources. For example, users may access the public cloud either based on *long-term reservations* or in an *on-demand* fashion [5]. In long-term reservation, a user first pays a one-time lump sum to reserve a certain amount of computing resources for a period of one or several years. Then, it gets charged extra for actually using the resource. While the extra payment may be lower than in on-demand services, it may not be desirable for each individual mobile user to pay for the reservation if the computing task load is light or moderate. *Brokers* have then been introduced to serve as an intermediate entity in favor of efficient cloud reservation [7], [8], [9]. Each broker reserves cloud resources for a set of multiple mobile users by exploiting the statistical characteristics of aggregate task flows. Then, the broker provides its users with cloud services at a resulting lower price, and schedules incoming computing tasks to the reserved clouds. The pricing strategy of computing services can become even more complex when cloudlets are incorporated into MCC networks. Cloudlets are in fact equipped with limited-only computational resources. Therefore, when the total computing requests exceed their capacity, sophisticated resource allocation policies should be invoked. Among these, dynamic-pricing-based policies such as auctioning allocate more resources to consumers that are willing to pay higher premiums for using resources [10], [11]; while fixed-pricing policies with dynamic admission control ensure efficient allocation of computing resources of cloudlets and fairness among consumers [12]. As a consequence, there is a need for each broker to optimize resource utilization to minimize its average cost by choosing between (i) reserving computing resources from the public cloud with fixed prices and (ii) from the cloudlet with adaptive prices while competing with other brokers. In MCC networks with multiple brokers, cloud reservation strategies with both architecture and pricing heterogeneities are, to the best of our knowledge, unexplored.

In this paper, we consider a MCC network composed of a public cloud, a cloudlet, and multiple brokers. Each broker tries to minimize the average price for its users to use the cloud services, by adjusting the amount of cloud resources from the public cloud, the cloudlet bidding strategy, and the task outsourcing strategy, while trying to meet predefined QoE constraints for the users. Under this setting, we study two classes of cloud reservation algorithms, *pure competition (PC)* and *compete-then-cooperate (CC)*. The main contributions of the paper are as follows.

- We consider a more complex scenario than in previous literature, where multiple brokers assign cloud resources to mobile users. The model is characterized by an heterogeneous cloud architecture and by

the heterogeneous pricing strategies of cloud service providers;

- We study a *purely competitive scenario* where brokers compete to reserve computing resources from remote public clouds (which are affected by long delays) and from local cloudlets (which have limited computational resources but short delays). We provide theoretical results demonstrating the existence of disagreement points (DP) (i.e., the equilibrium reservation strategy that no broker has incentive to deviate unilaterally from) and convergence of the best-response strategies of the brokers to disagreement points;
- We study a *cooperative scenario* and formulate a problem where the objective is to minimize the total average price of all brokers, under the constraint that no broker should pay a price higher than the disagreement price (i.e., the competitive price);
- We design new globally optimal solution algorithm to solve the resulting non-convex cooperative problem, based on a combination of the *branch and bound framework* and of *advanced convex relaxation techniques*;
- We show that considerable cooperative gain can be achieved over pure competition in markets with only few brokers, and highlight the fact that the **cooperative gain is low in crowded markets**, i.e., with a high number of brokers.

The rest of the paper is organized as follows. Related work is discussed in Section 2. In Section 3, we describe the system model and problem formulation; In Section 4 we present the disagreement analysis, and in Section 5 we present the centralized optimization solution algorithm. We present simulation results and analysis in Section 6, and finally in Section 7 we draw some conclusions.

## 2 RELATED WORK

Cooperative resource allocation in cloud computing networks has recently attracted significant attention [7], [13], [14], [15]. For example, in [7], Wu et al. investigate a queueing-based video-on-Demand (VoD) systems and propose a dynamic cloud resource provisioning algorithm that can effectively support VoD streaming with low cloud utilization cost. Similarly, in [8], Niu et al. investigate a pricing-based bandwidth reservation scheme, and propose to use a profit-making broker to control the performance risks by multiplexing the bandwidth reservations of multiple VoD providers. In [13], the authors propose a semi-Markov-based decision making system for inter-domain service transfer to balance the computation loads among multiple cloud domains, with the objective of minimizing the number of service rejections that degrade the user satisfaction level significantly. In [14], Niyato et al. focus on a multi-organization cloud computing system, and study the problem of coalition formation considering the challenges of virtual machine management. Different from these works, where their primary focus is on homogeneous cloud pricing strategies, in this paper we consider *heterogeneous pricing* strategies.

Heterogeneous pricing strategies have also been studied in existing works on cloud resource allocation. For example,

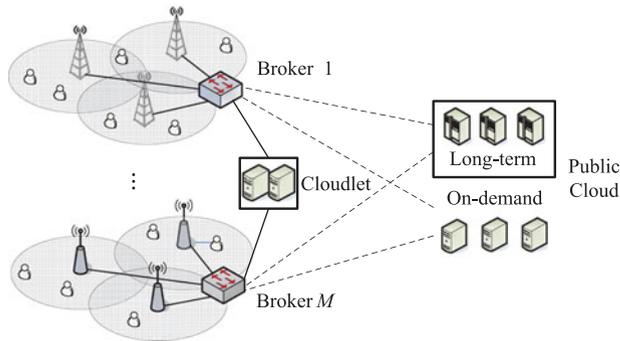


Fig. 1. Illustration of the MCC network with public cloud and cloudlet.

in [16], Chaisiri et al. study the problem of cost minimization for provisioning virtual servers in Amazon EC2 Cloud systems. In [17], Wang et al. consider an EC2-like pricing scheme with traditional pay-as-you-go pricing augmented by an auction market, and study the arising optimal capacity segmentation problem. In [18], Teng and Magoulès proposes a new resource pricing and allocation policy in which users are enabled to predict the future resource price as well as satisfy budget and deadline constraints. In addition to heterogeneous pricing strategies, in this work we also consider *heterogeneous cloud architectures*.

Recent literature has also focused on competition-based resource allocation. In [19], Tsai and Tsai investigate a bid-proportional auction scheme for resource allocation in capacity-constrained clouds. In [20], the authors propose a fully distributed VM-multiplexing resource allocation scheme to manage decentralized resources, with the objective of maximizing the resource utilization using the proportional share model (PSM) and delivering provably optimal execution efficiency. While these contributions focus on homogeneous pricing strategies, we study both competitive and cooperative behaviors in *MCC networks with multiple brokers*.

The main novel contribution of this paper lies therefore in the formulation and analysis of multi-broker MCC networks with both heterogeneous cloud architectures and pricing strategies. To the best of our knowledge, this is the first work presenting theoretical results about competitive and cooperative behaviors in MCC networks by jointly considering the hybrid architecture heterogeneity of the network and the statistical traffic characteristics offered by coexisting brokers.

### 3 SYSTEM MODEL

We consider a MCC network that consists of (i) one public cloud, (ii) one cloudlet, and (iii) a set  $\mathcal{M}$  with  $|\mathcal{M}| = M$  of brokers, as shown in Fig. 1. Mobile users generate computing tasks, and offload and run the tasks to the cloud server through the corresponding brokers. Each broker reserves cloud resources from both the public cloud and the cloudlet, and then offers the reserved resources to its users. The objective of each broker is to *minimize the average payment of its users while meeting the QoE constraints (expressed in terms of average computation delay)*. In practical networks, brokers can be dedicated and non-dedicated: dedicated brokers make revenue by reserving cloud resources for a group of mobile users at a lower price and then selling the reserved resources at a

higher price; non-dedicated brokers provide lower-price cloud resources to its served users as a side benefit. An example of non-dedicated brokers is the base stations operated by the same or different mobile service providers (MSPs); the MSPs have clear incentives to serve as brokers since by lowering the average cost of using cloud resources mobile users may become more likely to offload their computing tasks, which will result in more data traffic hence higher revenue for the MSPs. How to model the “willingness” of mobile users to offload their tasks will be studied in our future work. As a side comment, practical MCC networks may consist of multiple public clouds and cloudlets, and even within one public cloud, servers can be deployed in locations with different network bandwidth. For analytical tractability, in this paper we consider the model illustrated in Fig. 1. Future extensions of our work will consider scenarios with multiple cloudlets, multiple public clouds, and public cloud with heterogeneous network bandwidth (which may not be theoretically tractable). Next, we describe the cloud model and the QoE model, respectively.

#### 3.1 Cloud Pricing Model

According to different resource constraints of the public cloud and of the cloudlet, several different pricing strategies can be applied, e.g., long-term reservation, on-demand request, and auction-based allocation. In this work, we assume that long-term reservation and on-demand request strategies are in place at the public cloud, while auction-based allocation policies are implemented at the cloudlet because of its potentially limited resources.<sup>1</sup>

*Public cloud pricing.* We consider as in [5], [17] a linear cost model for the public cloud. Denote by  $p^L$  and  $p^D$  the price of the cloud resources corresponding to long-term reservation and on-demand request (as introduced in Section 1), in terms of  $\$/\text{VH}$ , i.e., the payment required to use one virtual machine (VM) for one hour. Then, the one-hour revenue of broker  $m \in \mathcal{M}$  for these two families of cloud resources, denoted by  $c_m^L(v_m^L)$  and  $c_m^D(v_m^D)$ , respectively, can be represented as

$$c_m^L(v_m^L) = p^L v_m^L, \quad m \in \mathcal{M}, \quad (1)$$

$$c_m^D(v_m^D) = p^D v_m^D, \quad m \in \mathcal{M}, \quad (2)$$

where  $v_m^L$  and  $v_m^D$  are the number of virtual machines (VMs) that broker  $m \in \mathcal{M}$  occupies in the long-term and on-demand manners, respectively.

*Cloudlet pricing.* Different auction schemes can be applied to obtain efficient resource allocation in the cloudlet, e.g., threshold-based policy [5], or bid proportion policy [18]. With the threshold-based policy, each user submits a bidding price, and is charged with the threshold price set by the cloud provider if the submitted price is higher than the threshold, and is denied access to cloud resources otherwise. Although this policy is adopted by some commercial

1. Cloudlets are usually operated by small organizations, like a campus or a coffee shop. While they usually provide free computing services to their students or customers, other users may still need to pay to use the cloud resources. Otherwise, if a cloudlet is completely freely accessible, it may be congested by too many submitted computing requests, causing the problem of “tragedy of the commons” [21].

cloud providers, like Amazon EC2 [5], it results in very complicated price model that is nonlinear and even non continuous, and hence not tractable. We consider a proportional policy, in which the portion of computing resources allocated to bidders is proportional to their purchasing price; it will be the subject of our future work to incorporate alternative auctioning models (e.g., as in [5]) and fixed pricing models with dynamic admission control (e.g., as in [12]).

Denote by  $\mathbf{p}^T = (p_m^T)_{m \in \mathcal{M}}$  the price vector of all brokers in  $\mathcal{M}$ . Then, the number of VMs allocated to broker  $m$ , denoted by  $v_m^T(\mathbf{p}^T)$ , can be represented as

$$v_m^T(\mathbf{p}^T) = v_0 \times \frac{p_m^T}{\sum_{n \in \mathcal{M}} p_n^T}, \quad \forall m \in \mathcal{M} \quad (3)$$

with

$$p_m^T \geq p_0, \quad \forall m \in \mathcal{M}, \quad (4)$$

where  $v_0$  represents the total number of VMs in the cloudlet, and  $p_0$  is the lowest price acceptable by the cloudlet. We assume that a VM can be shared by multiple brokers, i.e.,  $v_m^T$  in (3) may take real values. Then, the one-hour payment of broker  $m$  to the cloudlet, denoted by  $c_m^T(\mathbf{p}^T)$ , can be expressed as

$$c_m^T(\mathbf{p}^T) = p_m^T v_m^T(\mathbf{p}^T), \quad m \in \mathcal{M}. \quad (5)$$

Finally, the total one-hour cost of broker  $m \in \mathcal{M}$ , denoted by  $c_m(v_m^D, \mathbf{p}^T)$  can be expressed as<sup>2</sup>

$$c_m(v_m^D, \mathbf{p}^T) = c_m^L(v_m^L) + c_m^D(v_m^D) + c_m^T(\mathbf{p}^T), \quad (6)$$

and the average cloud price, denoted by  $p_m$  for broker  $m \in \mathcal{M}$ , is then

$$p_m(v_m^D, \mathbf{p}^T) = \frac{c_m(v_m^D, \mathbf{p}^T)}{v_m^L + v_m^D + v_m^T(\mathbf{p}^T)}. \quad (7)$$

### 3.2 Quality of Experience Model

In addition to the average cost for occupying the cloud resources, another important QoE metric is the average delay between the time when a user submits a computing request and the time when it gets back the computing result from the cloud. To characterize the delay behavior of the MCC network, we assume that computing tasks are generated, for all mobile users served by broker  $m \in \mathcal{M}$ , following a Poisson process with mean  $A_m^0$ . As in [22], we further assume the computational complexity of each task to be exponentially distributed with mean  $\mu_m$ , measured in number of CPU FLOPs.

Computing tasks can be served in different ways in each cloud. In the case of no VM multiplexing, i.e., each VM serves one task at a time, the cloud system can be modeled as an  $M/M/k$  queue. In the case of full multiplexing, i.e., the task being served occupies all the computing resources through parallel computing, the system can be approximately characterized using an  $M/M/1$  queue if the intra-

cloud delay to split a task among servers is negligible only. Otherwise,  $M/M/1$  can be extended to  $M/G/1$  queue to account for the non-negligible intra-cloud delay; in this case, it can be hard to derive closed-form expressions for the average sojourn time (including waiting and service time) of computing tasks because of generally distributed service time (as indicated by  $G$  in  $M/G/1$ ) [23]. In this work, we consider the case without VM multiplexing as a lower-bound on the system performance, while leaving the other case (and their combinations) for future work. Furthermore, in order to guarantee an acceptable level of QoE for a various set of multimedia services, like photo or video editing [6], we limit the average waiting time of tasks in each cloud to be at a low level, implying a light-load queue at each cloud. Then, the average sojourn time, denoted by  $T_m^L$ ,  $T_m^D$  and  $T_m^T$  for the long-term reserved cloud, on-demand requested cloud, and the cloudlet, respectively, can be represented as [23]

$$T_m^L(v_m^L, A_m^L) = \frac{v_m^L}{\mu_m v_m^L - A_m^L}, \quad \forall m \in \mathcal{M}, \quad (8)$$

$$T_m^D(v_m^D, A_m^D) = \frac{v_m^D}{\mu_m v_m^D - A_m^D}, \quad \forall m \in \mathcal{M}, \quad (9)$$

$$T_m^T(v_m^T(\mathbf{p}^T), A_m^T) = \frac{v_m^T(\mathbf{p}^T)}{\mu_m v_m^T(\mathbf{p}^T) - A_m^T}, \quad \forall m \in \mathcal{M}, \quad (10)$$

where  $A_m^L$ ,  $A_m^D$  and  $A_m^T$  represent the average task rate that broker  $m \in \mathcal{M}$  outsources to the long-term reserved cloud, on-demand requested cloud, and the cloudlet, respectively;  $\mu_0$  represents the number of CPU FLOPs available to each VM per second. Then, we have

$$A_m^L + A_m^D + A_m^T = A_m^0, \quad \forall m \in \mathcal{M}, \quad (11)$$

and the average delay constraint can then be expressed as

$$T_m^{U2B} + T_m^{B2P} + T_m^L(v_m^L, A_m^L) \leq T_m^0, \quad \forall m \in \mathcal{M}, \quad (12)$$

$$T_m^{U2B} + T_m^{B2P} + T_m^D(v_m^D, A_m^D) \leq T_m^0, \quad \forall m \in \mathcal{M}, \quad (13)$$

$$T_m^{U2B} + T_m^{B2T} + T_m^T(v_m^T(\mathbf{p}^T), A_m^T) \leq T_m^0, \quad \forall m \in \mathcal{M}, \quad (14)$$

with  $T_m^0$  as the average delay acceptable by users served by broker  $m \in \mathcal{M}$ , and  $T_m^{U2B}$ ,  $T_m^{B2T}$  and  $T_m^{B2P}$  represent the average time to transmit a task from user to broker, from broker to cloudlet, and from broker to public cloud, respectively. Again, to highlight the effects of the cloud reservation strategy on the QoE of mobile users, we consider  $T_m^{U2B}$ ,  $T_m^{B2T}$  and  $T_m^{B2P}$  to be constant and known.

### 3.3 Problem Statement

From a global system perspective, an appealing design objective is to find the allocation strategy for all MSPs that minimizes the average cost of all users using the cloud service, by cooperatively deciding the cloud reservation strategies and task outsourcing strategies of all the brokers. However, different brokers may be run by different organizations, and may therefore be selfish and only willing to minimize their individual prices. In other words, there is no incentive for the brokers to cooperate if the resulting price is not lower compared with that achievable through pure

2. We do not express  $c_m$  as a function of  $v_m^L$ , since it is not easy for brokers to change their long-term reservation strategies. Hence,  $v_m^L$  is considered to be fixed in the following analysis.

competition. We call the latter disagreement price or disagreement point. Let  $p_m^{\text{DP}}$  represent a disagreement point price for broker  $m \in \mathcal{M}$ . Then, a possible precondition (alternative relaxed preconditions are discussed in Section 5.3) for the brokers to cooperate can be expressed as

$$p_m(v_m^{\text{D}}, \mathbf{p}^{\text{T}}) \leq p_m^{\text{DP}}, \forall m \in \mathcal{M}. \quad (15)$$

The ‘‘social’’ problem can then be formulated as<sup>3</sup>

$$\begin{aligned} \text{Given :} & \quad p_m^{\text{L}}, v_m^{\text{L}}, A_m^0, \forall m \in \mathcal{M} \\ \text{Find :} & \quad \mathbf{p}^{\text{T}}, v_m^{\text{D}}, A_m^{\text{L}}, A_m^{\text{D}}, A_m^{\text{T}}, \forall m \in \mathcal{M} \\ \text{Minimize :} & \quad U = \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} p_m(v_m^{\text{D}}, \mathbf{p}^{\text{T}}) \\ \text{subject to :} & \quad (4), (11), (12), (13), (14), (15), \end{aligned} \quad (16)$$

where  $U$  represents the average price of all brokers achievable through cooperative cloud reservation, constraint (4) defines the pricing strategy domain for all brokers, (11)-(14) define the task outsourcing strategies, and (15) is the disagreement price constraint used to ensure that the resulting cost can be further lowered for all brokers compared with that of pure competition.

In the following, we try to provide an answer to the following related questions: (i) Does such a disagreement price exist for a given system setting? If it does, how can we determine it? (ii) How much cooperative gain can be achieved, i.e., what is the value of cooperation? Is it necessary for brokers to cooperate? Next, we investigate the two questions in Section 4 and Section 5, respectively.

## 4 DISAGREEMENT POINT ANALYSIS

Disagreement point is a notion from cooperative game theory [24], representing the lowest utility for players (i.e., brokers) involved in a cooperative game to have incentive to cooperate. In a multi-entity market, a potential approach to determine the DP is to rely on a pure competition strategy. In this case, the resulting DP (if it exists) corresponds to a Nash equilibrium (NE) of the competitive market [24]. Next, we first present a formal definition of DP, and then study its existence and achievability.

**Definition 1 (Disagreement Point).** Denoting by  $\mathbf{p}_{-m}^{\text{T}} = (p_n^{\text{T}})_{n \in \mathcal{M}/m}$  the bidding price vector of all brokers in  $\mathcal{M}$  except  $m$ , the average price  $p_m(v_m^{\text{D}}, \mathbf{p}^{\text{T}})$  in (7) can be rewritten as  $p_m(v_m^{\text{D}}, p_m^{\text{T}}, \mathbf{p}_{-m}^{\text{T}})$  for each  $m \in \mathcal{M}$ . Then,  $\mathbf{p}^* \triangleq [p_m((v_m^{\text{D}})^*, (p_m^{\text{T}})^*, (\mathbf{p}_{-m}^{\text{T}})^*))]_{m \in \mathcal{M}}$  is called a disagreement point if the following conditions can be simultaneously satisfied for all  $m \in \mathcal{M}$ :

$$((v_m^{\text{D}})^*, (p_m^{\text{T}})^*) \triangleq \arg \min_{(v_m^{\text{D}}, p_m^{\text{T}}) \in \Phi_m} p_m(v_m^{\text{D}}, p_m^{\text{T}}, (\mathbf{p}_{-m}^{\text{T}})^*), \quad (17)$$

with  $\Phi_m$  being the domain set of broker  $m$  defined through constraints (4), (8)-(14).

3. The social problem can also be formulated as a weighted sum-utility maximization problem, e.g., assign different weights to the brokers by considering the number of end users served by each of them. This does however not change essentially the following disagreement point analysis in Section 4 and DP-constrained cost minimization in Section 5.

### 4.1 Existence of DP

Assume that time-sharing among different brokers is enabled in each cloud for occupying the computing resources, meaning that in each competition period, a broker is allowed to reserve fractional number of VMs. Then, we have following theorem:

**Theorem 1.** *There exists at least one disagreement point (as in Definition 1) for the problem formulated in (16).*

**Proof.** Since we define the DP based on the pure competitive market, the individual optimization problem for each broker  $m \in \mathcal{M}$  can be written as

$$\begin{aligned} \text{Given :} & \quad p_{-m}^{\text{T}}, v_m^{\text{L}}, A_m^0 \\ \text{Find :} & \quad p_m^{\text{T}}, v_m^{\text{D}}, A_m^{\text{L}}, A_m^{\text{D}}, A_m^{\text{T}} \\ \text{Minimize :} & \quad p_m(v_m^{\text{D}}, p_m^{\text{T}}, \mathbf{p}_{-m}^{\text{T}}) \\ \text{subject to :} & \quad (4), (11), (12), (13), (14). \end{aligned} \quad (18)$$

Then, to prove the theorem, we only need to transform the problem formulated in (18), and show that the theorem holds true for the resulting equivalent problem.

The transformation is based on the observation from (8)-(10) that, the sojourn time of tasks in each cloud is a monotonically decreasing function with the number of the available VMs. This implies that, when  $p_m(v_m^{\text{D}}, p_m^{\text{T}}, \mathbf{p}_{-m}^{\text{T}})$  in (18) is minimized, the equality holds for constraints (12), (13), (14). Then, we have

$$T_m^{\text{L}}(v_m^{\text{L}}, A_m^{\text{L}}) = T_m^0 - T_m^{\text{U2B}} - T_m^{\text{B2P}}, \quad (19)$$

$$T_m^{\text{D}}(v_m^{\text{D}}, A_m^{\text{D}}) = T_m^0 - T_m^{\text{U2B}} - T_m^{\text{B2P}}, \quad (20)$$

$$T_m^{\text{T}}(v_m^{\text{T}}, \mathbf{p}_{-m}^{\text{T}}, A_m^{\text{T}}) = T_m^0 - T_m^{\text{U2B}} - T_m^{\text{B2T}}. \quad (21)$$

Then, together with (8), (9) and (10), we can express the outsourcing task rate  $A_m^{\text{L}}$ ,  $A_m^{\text{D}}$  and  $A_m^{\text{T}}$  as a function of the corresponding number of available VMs in each cloud, as follows:

$$A_m^{\text{L}}(v_m^{\text{L}}) = \left( \frac{\mu_0}{\mu_m} - \frac{1}{T_m^{\text{L}}} \right) v_m^{\text{L}}, \quad (22)$$

$$A_m^{\text{D}}(v_m^{\text{D}}) = \left( \frac{\mu_0}{\mu_m} - \frac{1}{T_m^{\text{D}}} \right) v_m^{\text{D}}, \quad (23)$$

$$A_m^{\text{T}}(v_m^{\text{T}}, \mathbf{p}_{-m}^{\text{T}}, \mathbf{p}_{-m}^{\text{T}}) = \left( \frac{\mu_0}{\mu_m} - \frac{1}{T_m^{\text{T}}} \right) v_m^{\text{T}}(p_m^{\text{T}}, \mathbf{p}_{-m}^{\text{T}}), \quad (24)$$

with  $T_m^{\text{L}}$ ,  $T_m^{\text{D}}$  and  $T_m^{\text{T}}$  defined in (19)-(21), respectively. Further, for a given number of long-term reserved VMs for each broker  $m \in \mathcal{M}$  and from constraint (11), we can express the number of on-demand requested VMs as

$$v_m^{\text{D}}(p_m^{\text{T}}, \mathbf{p}_{-m}^{\text{T}}) = \frac{A_m^0 - A_m^{\text{T}}(p_m^{\text{T}}, \mathbf{p}_{-m}^{\text{T}}) - A_m^{\text{L}}(v_m^{\text{L}})}{\frac{\mu_0}{\mu_m} - \frac{1}{T_m^{\text{D}}}}. \quad (25)$$

Then, we can represent the average price  $p_m(v_m^{\text{D}}, p_m^{\text{T}}, \mathbf{p}_{-m}^{\text{T}})$  in (7) as a function of only  $(p_m^{\text{T}}, \mathbf{p}_{-m}^{\text{T}})$ , as follows:

$$p_m(p_m^T, \mathbf{p}_{-m}^T) = \frac{p_m^L v_m^L + p_m^D v_m^D(p_m^T, \mathbf{p}_{-m}^T) + p_m^T v_m^T(p_m^T, \mathbf{p}_{-m}^T)}{v_m^L + v_m^D(p_m^T, \mathbf{p}_{-m}^T) + v_m^T(p_m^T, \mathbf{p}_{-m}^T)}, \quad (26)$$

with  $v_m^D(p_m^T, \mathbf{p}_{-m}^T)$  and  $v_m^T(p_m^T, \mathbf{p}_{-m}^T)$  defined in (25) and (3), respectively. Finally, the optimization problem (18) can be transformed into the following problem:

$$\begin{aligned} \text{Given :} & \quad \mathbf{p}_{-m}^T \\ \text{Minimize :} & \quad p_m(p_m^T, \mathbf{p}_{-m}^T), \\ & \quad p_m^T \in \widehat{\Phi}_m \end{aligned} \quad (27)$$

where  $\widehat{\Phi}_m$  is the domain set of broker  $m \in \mathcal{M}$  defined through constraint (4); the problem is convex as formalized in Theorem 2, completing the proof according to the existence results in [25, P39].  $\square$

**Theorem 2.** *The optimization problem (27) is a convex optimization problem.*

**Proof.** We first show that  $p_m(p_m^T, \mathbf{p}_{-m}^T)$  is either convex or concave. For this purpose, we transform it as follows. Denote  $\frac{\mu_0}{\mu_m} - \frac{1}{T_m^L}$ ,  $\frac{\mu_0}{\mu_m} - \frac{1}{T_m^D}$ , and  $\frac{\mu_0}{\mu_m} - \frac{1}{T_m^T}$  in (22), (23) and (24) as  $\alpha_m^L$ ,  $\alpha_m^D$  and  $\alpha_m^T$ , respectively. Then, the average cloud price  $p_m(p_m^T, \mathbf{p}_{-m}^T)$  in (26) can be rewritten as

$$p_m(p_m^T, \mathbf{p}_{-m}^T) = \frac{p_m^L v_m^L + \frac{p_m^D}{\alpha_m^D} [A_m^0 - A_m^T(\mathbf{p}^T) - A_m^L(v_m^L)] + p_m^T v_m^T(\mathbf{p}^T)}{v_m^L + \frac{A_m^0 - A_m^T(\mathbf{p}^T) - A_m^L(v_m^L)}{\alpha_m^D} + v_m^T(\mathbf{p}^T)}, \quad (28)$$

which can be further rewritten as

$$p_m(p_m^T, \mathbf{p}_{-m}^T) = \frac{(p_m^T)^2 + \beta_m^T p_m^T + \gamma_m^T \sum_{n \in \mathcal{M}/m} p_n^T}{\psi_m^T p_m^T + \phi_m^T \sum_{n \in \mathcal{M}/m} p_n^T}, \quad (29)$$

with

$$\beta_m^T = \frac{[p_m^L v_m^L + \frac{p_m^D}{\alpha_m^D} (A_m^0 - A_m^L)]}{v_0} - \frac{\alpha_m^T}{\alpha_m^D} p_m^D, \quad (30)$$

$$\gamma_m^T = \frac{[p_m^L v_m^L + \frac{p_m^D}{\alpha_m^D} (A_m^0 - A_m^L)]}{v_0}, \quad (31)$$

$$\psi_m^T = 1 - \frac{\alpha_m^T}{\alpha_m^D} + \frac{v_m^L + \frac{A_m^0 - A_m^L}{\alpha_m^D}}{v_0}, \quad (32)$$

$$\phi_m^T = \frac{v_m^L + \frac{A_m^0 - A_m^L}{\alpha_m^D}}{v_0}. \quad (33)$$

Finally, we can rewrite  $p_m(p_m^T, \mathbf{p}_{-m}^T)$  in the following form:

$$p_m(p_m^T, \mathbf{p}_{-m}^T) = \frac{1}{\psi_m^T} \left[ p_m^T + \beta_m^T - \frac{\phi_m^T}{\psi_m^T} \sum_{n \in \mathcal{M}/m} p_n^T + \frac{\left( \frac{\phi_m^T}{\psi_m^T} \sum_{n \in \mathcal{M}/m} p_n^T \right)^2 + \left( \gamma_m^T - \beta_m^T \frac{\phi_m^T}{\psi_m^T} \right) \sum_{n \in \mathcal{M}/m} p_n^T}{p_m^T + \frac{\phi_m^T}{\psi_m^T} \sum_{n \in \mathcal{M}/m} p_n^T} \right], \quad (34)$$

which is either convex or concave with respect to  $p_m^T$  with given  $\mathbf{p}_{-m}^T$  for any  $m \in \mathcal{M}$ .

It can be verified that,  $p_m(p_m^T, \mathbf{p}_{-m}^T)$  decreases with  $p_m^T$  if  $p_m^T$  is close to zero, while increases when  $p_m^T$  is sufficiently large, and this implies the convexity of  $p_m(p_m^T, \mathbf{p}_{-m}^T)$  with respect to  $p_m^T$ .  $\square$

## 4.2 Algorithm

Assuming a pure competitive market, we adopt a best-response algorithm to iteratively achieve an approximation of the disagreement point. A Jacobi version of the best-response algorithm is given in Algorithm 1, where  $p_m(p_m^T, (\mathbf{p}_{-m}^T)^\kappa)$  in (35) is given in (26). A convergence property of the algorithm is given in Theorem 3 below.

### Algorithm 1. Jacobi Best-response Algorithm

**Data :**  $(\mathbf{p}^T)^0 \in \Phi$ . Set  $\kappa = 0$ .

(S.1) : If  $(\mathbf{p}^T)^\kappa$  satisfies a suitable termination criterion: STOP;

(S.2) : Each broker  $m \in \mathcal{M}$  computes its best-response:

$$(p_m^T)^{\kappa+1} \in \arg \min_{p_m^T \in \widehat{\Phi}_m} p_m(p_m^T, (\mathbf{p}_{-m}^T)^\kappa) \quad (35)$$

(S.3) :  $\kappa \leftarrow \kappa + 1$  and go to (S.1).

**Theorem 3 (Convergence of Algorithm 1).** *In the setting described above, Algorithm 1 is guaranteed to converge to a DP of the problem formulated in (16) if the number of brokers is sufficiently large, i.e., with large  $|\mathcal{M}|$ .*

**Proof.** We first obtain the first derivative of  $p_m(p_m^T, \mathbf{p}_{-m}^T)$  with respect to  $p_m^T$  as

$$\frac{\partial p_m(p_m^T, \mathbf{p}_{-m}^T)}{\partial p_m^T} = \frac{1}{\psi_m^T} \left[ 1 - \frac{\left( \frac{\phi_m^T}{\psi_m^T} \sum_{n \in \mathcal{M}/m} p_n^T \right)^2 + \left( \gamma_m^T - \beta_m^T \frac{\phi_m^T}{\psi_m^T} \right) \sum_{n \in \mathcal{M}/m} p_n^T}{\left( p_m^T + \frac{\phi_m^T}{\psi_m^T} \sum_{n \in \mathcal{M}/m} p_n^T \right)^2} \right], \quad (36)$$

with  $\beta_m^T$ ,  $\gamma_m^T$ ,  $\psi_m^T$  and  $\phi_m^T$  defined in (30)-(33). Letting  $\frac{\partial p_m(p_m^T, \mathbf{p}_{-m}^T)}{\partial p_m^T} = 0$ , we have

$$\left( \frac{\phi_m^T}{\psi_m^T} \sum_{n \in \mathcal{M}/m} p_n^T \right)^2 + \left( \gamma_m^T - \beta_m^T \frac{\phi_m^T}{\psi_m^T} \right) \sum_{n \in \mathcal{M}/m} p_n^T = \left( p_m^T + \frac{\phi_m^T}{\psi_m^T} \sum_{n \in \mathcal{M}/m} p_n^T \right)^2 \quad (37)$$

$$\Rightarrow \left( \gamma_m^T - \beta_m^T \frac{\phi_m^T}{\psi_m^T} \right) = \frac{(p_m^T)^2}{\sum_{n \in \mathcal{M}/m} p_n^T} + 2p_m^T \frac{\phi_m^T}{\psi_m^T}. \quad (38)$$

When the number of brokers is large, we have  $\frac{(p_m^T)^2}{\sum_{n \in \mathcal{M}/m} p_n^T} \approx 0$ . Then, we have

$$p_m^T = \frac{1}{2} \left( \frac{\psi_m^T}{\phi_m^T} \gamma_m^T - \beta_m^T \right), \quad (39)$$

which implies that the best response of each broker depends only on the network setting through  $\beta_m^T, \gamma_m^T, \psi_m^T$  and  $\phi_m^T$ , and also implies the convergence of the best-response algorithm.  $\square$

The theorem provides a sufficient condition for Algorithm 1 to converge to a DP. In practical MCC networks, the cloud price is usually much smaller than 1 in \$/VH [5], the item  $\frac{(p_m^T)^2}{\sum_{n \in \mathcal{M}/m} p_n^T}$  can be close to 0 with only a few brokers, e.g., 4 as shown in Fig. 7. In the case of fewer brokers, hence non-negligible  $\frac{(p_m^T)^2}{\sum_{n \in \mathcal{M}/m} p_n^T}$ , we can approximate the DP by averaging the resulting best-response bidding prices over a larger number of iterations.

### 4.3 Implementation Issues

In each iteration of Algorithm 1, to minimize the cost  $p_m(p_m^T, \mathbf{p}_{-m}^T)$  defined in (26), each broker  $m \in \mathcal{M}$  needs to gather the bidding information of all other brokers in  $\mathcal{M}/m$ , i.e.,  $\mathbf{p}_{-m}^T$ . From (3), (24), (25), and (26) we see that  $p_m(p_m^T, \mathbf{p}_{-m}^T)$  depends on the sum of  $\mathbf{p}_{-m}^T$  only. This implies that broker  $m$  does not need to gather the individual bidding information of other brokers, and instead it just needs to infer, according to the adopted cloudlet pricing policy in (3), the sum  $\mathbf{p}_{-m}^T$  from (i) its own bidding price and (ii) the amount of cloud resources that are eventually assigned to it. Hence, the algorithm can be implemented in a fully distributed manner that does not require sharing of any information among brokers. Moreover, while Algorithm 1 presents a Jacobi version of the best-response algorithm, it can also be implemented in Gauss-Seidel manner, and even in a fully asynchronous fashion [26] with convergence guarantee in the setting discussed in Theorem 3.

It is worth pointing out that additional factors can also be incorporated into the cloudlet pricing strategies, e.g., by considering the heterogeneity of the workload of brokers, or considering the mobility of end users and their willingness to offload their computing tasks given a pricing strategy; this however may result in quite different (usually more complex) optimization problems, with different expressions of (3), (24), (25), and (26). Moreover, in those cases information sharing may be necessary to attain a disagreement point among the brokers. Therefore, it may also become essential to design schemes ensuring trust and enforcing truthfulness, and to analyze the effects of untruthfulness on each broker's achievable utility as well as existence and attainability of the disagreement point. These will be topics of future research.

## 5 DP CONSTRAINED COST MINIMIZATION

After having derived the disagreement point in the previous section, we now focus on the social problem formulated in (16), i.e., minimizing the average price of all brokers while guaranteeing that their individual price is not higher than that corresponding to the disagreement point. By substituting the transformed average price (26) into (16), the social problem can be rewritten as

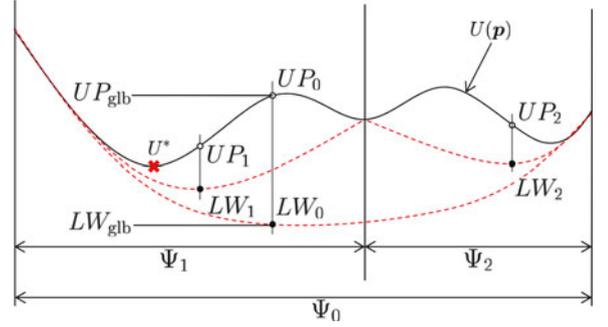


Fig. 2. Illustration of the globally optimal solution algorithm.

$$\begin{aligned} \text{Given :} & \quad p_m^L, v_m^L, m \in \mathcal{M} \\ \text{Minimize :} & \quad U = \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} p_m(\mathbf{p}^T) \quad (40) \\ \text{subject to :} & \quad (4), (11), (12), (13), (14), (15), \end{aligned}$$

where  $\Psi \triangleq \prod_{m \in \mathcal{M}} \hat{\Phi}_m$  denotes the joint domain set of all brokers in  $\mathcal{M}$ . Although the individual optimization problem discussed in the previous section is convex, the social problem is nonlinear and nonconvex due to the coupled bidding strategies when competing for the cloudlet resources. We therefore derive a non-heuristic globally optimal solution algorithm.

### 5.1 Overview of the Solution Algorithm

The algorithm is designed based on a combination of the branch and bound framework and of convex relaxation techniques [27] to obtain a globally optimal solution with a predefined optimality precision  $\varepsilon \in (0, 1]$  which can be arbitrarily close to 1. Fig. 2 shows an illustration of the proposed algorithm. Denoting the globally optimal objective function in (40) as  $U^*$ , then the objective of the algorithm is to iteratively search for a  $U$  satisfying  $U \leq U^*/\varepsilon$ .

For this purpose, the algorithm iterates by maintaining a global upper and global lower bounds on the objective function  $U$  in (40), denoted by  $UP_{\text{glb}}$  and  $LW_{\text{glb}}$ , respectively. Then,

$$LW_{\text{glb}} \leq U^* \leq UP_{\text{glb}}. \quad (41)$$

In addition to this, the algorithm also maintains a set  $\tilde{\Psi}$  of sub-domains that is initialized to be  $\tilde{\Psi} = \{\Psi_0 = \Psi\}$  with  $\Psi$  defined in (40). As the iterations proceed, the algorithm partitions  $\Phi_0$  into a series of sub-domains  $\tilde{\Psi} = \{\Psi_i \subset \Psi_0, i = 1, 2, \dots\}$ . For each  $\Psi_i$ , the algorithm obtains a local upper and lower bound on  $U$  over the domain (specific methods are discussed later), denoted by  $UP(\Psi_i)$  and  $LW(\Psi_i)$ , and updates the global upper and lower bounds as follows:

$$UP_{\text{glb}} = \min_i \{UP(\Psi_i)\}, \quad (42)$$

$$LW_{\text{glb}} = \min_i \{LW(\Psi_i)\}. \quad (43)$$

For example, the global bounds are updated to be  $UP_{\text{glb}} = UP(\Psi_0)$  and  $LW_{\text{glb}} = LW(\Psi_0)$ , respectively, after the first iteration, as shown in Fig. 2 with  $UP_0$  and  $LW_0$ ; and then updated to  $UP_1$  and  $LW_1$ , respectively, after the second iteration.

The iteration terminates if  $UP_{\text{glob}} \leq LW_{\text{glob}}/\varepsilon$  and the algorithm then sets the optimal objective  $U$  in (40) to  $U^* = UP_{\text{glob}}$ ; and otherwise, the algorithm chooses one sub-domain from  $\tilde{\Psi}$  and further partitions it into two sub-domains, calculates  $UP(\cdot)$  and  $LW(\cdot)$ , and updates the  $UP_{\text{glob}}$  and  $LW_{\text{glob}}$  as in (42) and (43). In our algorithm, we select the  $\Psi_i \in \tilde{\Psi}$  with the lowest local upper bound, i.e.,  $i = \arg \min_i LW(\Psi_i)$ , and partitions the domain into two sub-domains by splitting the bidding price variable  $p_m^T$  that has the greatest range from its middle. Based on the update criterion of  $UP_{\text{glob}}$  and  $LW_{\text{glob}}$  in (42) and (43), the gap between the two global bounds converges to 0 as the domain-partition progresses. Furthermore, from (41),  $UP_{\text{glob}}$  and  $LW_{\text{glob}}$  converge to the globally maximal objective function  $U^*$ .

## 5.2 Local Lower and Upper Bounds

At each iteration, we rely on a convex relaxation technique to obtain a local lower bound for the subproblem corresponding to the selected sub-domain  $\Psi_i$ , and then obtain a local upper bound through local search.

*Convex relaxation.* As shown in Section 4, the individual utility function  $p_m(p_m^T, \mathbf{p}_{-m}^T)$  in (26) can be written in the following form:

$$p_m(p_m^T, \mathbf{p}_{-m}^T) = \frac{(p_m^T)^2 + \beta_m^T p_m^T + \gamma_m^T \sum_{n \in \mathcal{M}/m} p_n^T}{\psi_m^T p_m^T + \phi_m^T \sum_{n \in \mathcal{M}/m} p_n^T} \quad (44)$$

$$= \frac{\gamma_m^T}{\phi_m^T} \left( 1 + \frac{\frac{(p_m^T)^2 + \beta_m^T p_m^T}{\gamma_m^T} - \frac{\psi_m^T}{\phi_m^T} p_m^T}{\frac{\psi_m^T}{\phi_m^T} p_m^T + \sum_{n \in \mathcal{M}/m} p_n^T} \right)$$

with  $\beta_m^T$ ,  $\gamma_m^T$ ,  $\psi_m^T$  and  $\phi_m^T$  being constant coefficients defined in Section 4. We notice from (44) that,  $p_m(p_m^T, \mathbf{p}_{-m}^T)$  is a monotonical function of  $\sum_{n \in \mathcal{M}/m} p_n^T$ , implying that a lower bound on  $p_m(p_m^T, \mathbf{p}_{-m}^T)$  can be obtained by setting all  $p_n^T$  with  $n \in \mathcal{M}/m$  either to their lower bounds over the current domain set  $\Psi_i$ , or to the upper bounds, i.e., setting  $p_m(p_m^T, \mathbf{p}_{-m}^T)$  to  $p_m(p_m^T, \tilde{\mathbf{p}}_{-m}^T)$  or  $p_m(p_m^T, \hat{\mathbf{p}}_{-m}^T)$ , with  $\tilde{\mathbf{p}}_{-m}^T$  and  $\hat{\mathbf{p}}_{-m}^T$  being the upper and lower bound of  $\mathbf{p}_{-m}^T$  over  $\Psi_i$ , respectively. In either case, the lower bound can be obtained easily by solving a convex optimization of  $p_m^T$ .

*Local search.* Let  $\mathbf{p}_{\text{rlx}}^T$  be the optimal bidding price vector obtained by solving the relaxed optimization problem, and denote  $p_{\text{rlx}}$  as the corresponding real average price calculated based on the original utility function (26). Then,  $\mathbf{p}_{\text{rlx}}$  is feasible if each individual price is lower than or equal to the disagreement point  $p_m^{\text{DP}}$  for broker  $m \in \mathcal{M}$ , and infeasible otherwise. In the latter case, the local upper bound is simply set to infinity.

## 5.3 Remarks

Recall that in Section 3.3, we considered the precondition in (15), i.e., the resulting cost through cooperation should not be greater than the price at the disagreement point. The precondition can also be relaxed by allowing collusion among brokers, i.e., they bid the cloud resources with the lowest cloudlet price  $p_0$  defined in (4), and then reallocate the resources later to achieve the best cost efficiency. This, however, may cause the cloudlet to invoke the self-

reservation protection [19]. Moreover, reallocating cloud resources requires brokers to forward computing tasks on behalf of each other if the cloudlet adopts strict resource access control for security, e.g., a virtual machine assigned to broker  $m$  can be accessed by that broker only. In real networks, it may not be easy to mutually forward computing tasks that can be potentially of large size (e.g., video transcoding), e.g., due to possibly high forwarding delay, protocol inconsistency or content security restrictions, especially if brokers are operated by different organizations. Finally, even if complete collusion is enabled, a fairness policy is still needed for reallocating resources among the brokers. Constraint (15) guarantees fairness in the sense that cooperating does not result in higher cost than pure competition for all brokers. In this work, we do not consider this kind of collusion-based cooperation.

## 6 SIMULATION RESULTS

We evaluate the performance of the considered two classes of algorithms, *pure competition (PC)* and *compete-then-cooperate (CC)*, and study the effects of different MCC network settings. A software-based simulator was implemented for both the competitive and the cooperative cloud reservation algorithms. The number of brokers is set to  $|\mathcal{M}| = 2, 4, \dots, 14$ . The total offered task rate, denoted by  $A_0$ , is uniformly distributed between [50 100], [150 200] and [200 250], and uniformly distributed among the brokers in each second. We consider an average delay threshold  $T_m^0 = 2s$ , which represents the upper-bound delay for acceptable QoE for a various set of applications [9], like multimedia compression and rendering, and non-interactive gaming, among others. A typical network setting is considered as follows. The average transmission delay is set to  $T_m^{\text{U2B}} = 100\text{ms}$  for links from the mobile users to each broker,  $T_m^{\text{B2T}} = 20\text{ms}$  from each broker to the cloudlet, and varied from 200 to 600 ms with step of 100 ms from each broker to the public cloud. The average computing time of tasks is set to  $\frac{\mu_m}{\mu_0} = 1$  for each broker, the number of long-term reserved VMs is set to 1.5 times of the average task incoming rate, the total number of VMs available in the cloudlet is set to  $v_0 = 50$ . The price of the on-demand public cloud is set to 0.06\$/VH corresponding to the default standard on-demand Amazon EC2 service [5], and 0.03\$/VH corresponding to a reservation service with a two-year plan. The price of the cloudlet is varied between [0.01 0.03]\$/VH.

All figures (except for the case study) are plotted by averaging over 50 independent simulation instances. In each simulation, the maximum number of iterations is set to 10 for the pure competition algorithm (i.e., Algorithm 1) and to 1,000 for the cooperative algorithm. The optimality precision  $\varepsilon$  in the cooperative algorithm is set to 0.95.

*Convergence.* The iteration procedure of the pure competition and the compete-then-cooperate algorithms are plotted in Figs. 3 and 4, respectively, by considering four brokers. We can see from Fig. 3 that the pure competition algorithm converges very fast, and from Fig. 4 that the predefined optimality precision (0.95 in our simulations) is achieved in around 700 iterations. Higher optimality precision can also be achieved at the cost of increased computational

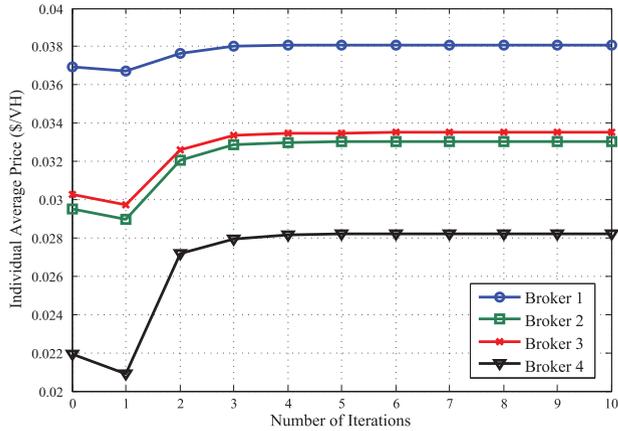


Fig. 3. Convergence of pure competition algorithm.

complexity, while the optimality precision can be reduced in favor of lower computational complexity in highly dynamic mobile environment so that brokers are able to update their cloud reservation quickly to adapt to the time-varying computing traffic load. As shown in Fig. 5, through 200 independent simulations, we found that an optimality precision higher than the predefined can be achieved in most cases (more than 90 percent of the tested instances). The probability that a level of optimality above the predefined threshold of 0.95 can be achieved through 1,000 iterations decreases with a higher number of brokers, e.g., 8, 10, or 12. However, we found that this is primarily because of the relaxation gap (introduced by the convex relaxation in Section 5.2), while the resulting feasible solution (obtained through local search) changes only slightly as iterations proceed. Hence, the adopted simulation setting is situated at a good tradeoff between computational complexity and optimality precision.

*Case study.* First, we study a simple case with two brokers to provide an intuitive understanding of the competitive and cooperative behaviors of brokers. In Fig. 6, we plot the price surfaces of the two brokers, from which the nonconvexity of the joint cloud reservation problem can be clearly observed. At the disagreement point, the bidding price that the two brokers submit to the cloudlet is 0.0339 and 0.0292, respectively, resulting in an average price of 0.0315. With cooperative reservation, the average price reduces to 0.0288 with bidding price 0.0206 and 0.0212. A price reduction ratio of 8.57 percent can be achieved,

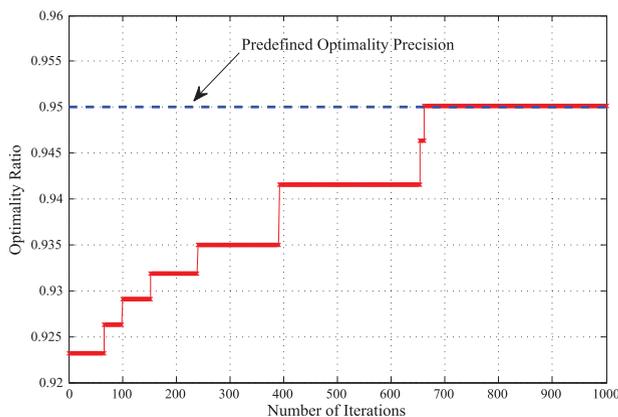


Fig. 4. Convergence of compete-then-cooperate algorithm.

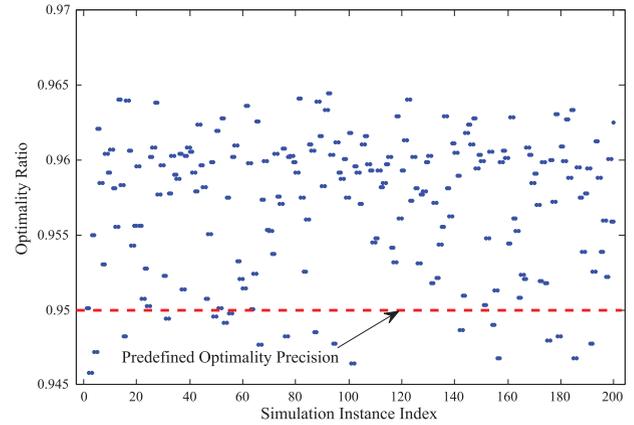


Fig. 5. Optimality precision achievable by the compete-then-cooperate algorithm.

with each broker's average price not greater than in pure competition. Several more examples are given in Table 1 with three brokers. We observe that price reduction can be achieved in all the tested instances.

*Average performance.* The effects of the number of brokers and of the traffic load are studied in Fig. 7. For comparison, we include a homogeneous scenario as a bottom line performance, where mobile users submit their computing tasks to the public cloud directly. We observe that a heterogeneous cloud architecture (i.e., with both public cloud and cloudlet) results in much lower average price compared with a homogeneous architecture with public cloud only. As indicated by the ellipses in Fig. 7, the price reduction achievable by pure competition is up to 23 percent compared to homogeneous cloud reservation, and as the number of brokers increases, the price reduction decreases, i.e., the price arises as the demand for resources increases. We also observe that the decreased speed in price reduction monotonically diminishes with the number of brokers and tends to become constant. For example, in the case of total offered task rate  $A_0 \in [50 \ 100]$ , the price reduction stays around 21.9 percent in a purely competitive market with more than 4 brokers. This verifies our statement in Section 4, i.e., the disagreement point price is independent of the number of brokers in a crowded competitive market. Compete-then-cooperate can achieve greater price reduction than pure competition,

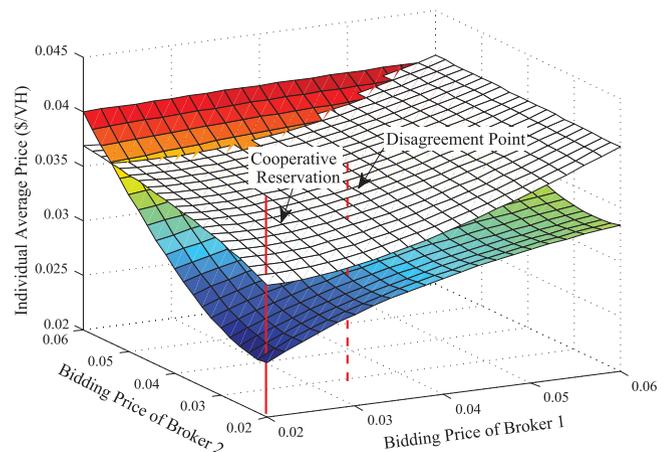


Fig. 6. Case study: Individual average price with two brokers.

TABLE 1  
Individual and Average Price Achievable by the PC and the CC Algorithms, and Price Reduction with Three Brokers

#		Individual		Average	Reduction
1	PC	0.0419	0.0349	0.0442	10.9%
	CC	0.0398	0.0254	0.0431	
2	PC	0.0407	0.0440	0.0412	7.4%
	CC	0.0367	0.0423	0.0376	
3	PC	0.0378	0.0441	0.0421	8.6%
	CC	0.0314	0.0426	0.0394	
4	PC	0.0424	0.0437	0.0401	7.3%
	CC	0.0396	0.0418	0.0356	
5	PC	0.0344	0.0434	0.0435	10.9%
	CC	0.0239	0.0420	0.0421	
6	PC	0.0441	0.0385	0.0418	8.43%
	CC	0.0426	0.0326	0.0389	
7	PC	0.0419	0.0442	0.0338	11.7%
	CC	0.0400	0.0433	0.0225	
8	PC	0.0377	0.0408	0.0444	9.0%
	CC	0.0316	0.0371	0.0430	
9	PC	0.0432	0.0427	0.0414	6.8%
	CC	0.0408	0.0401	0.0378	
10	PC	0.0426	0.0404	0.0435	7.1%
	CC	0.0399	0.0362	0.0415	

i.e., 30 percent with two brokers, and tends to 23.4 percent with more than 8 brokers. Comparing pure competition and compete-then-cooperate, we find that the benefits of cooperation in terms of price reduction vary from 8 to 1.9 percent as the number of brokers increases from 2 to 14. This means that cooperating in cloud reservation is more beneficial in less crowded markets, while the benefit is only marginal if the market becomes crowded. Similar results can be observed as the total offered task rate  $A_0$  becomes higher. For example, in case of  $A_0$  between [200 250], the average price changes only slightly in both the competitive and cooperative cases. In this case, the price reduction through cooperation is only very limited, less than 3 percent. Therefore, from this experiment we conclude that a cooperative

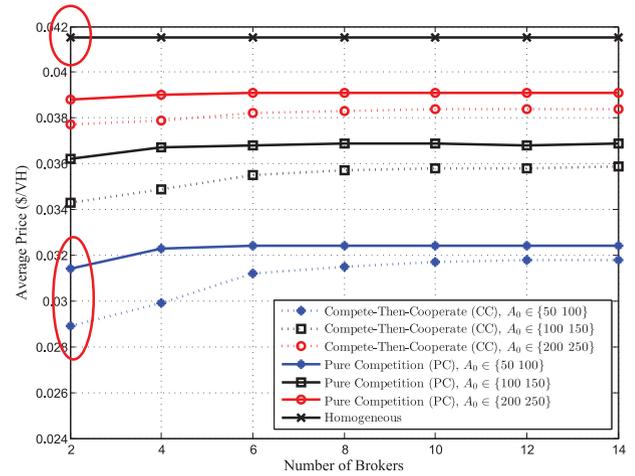


Fig. 7. Average price with varying number of brokers for different total offered task rates.

pricing scheme is only desirable in case of low number of brokers with light traffic levels.

In Fig. 8, the effects of the average transmission delay from the brokers to the public cloud on the competitive and cooperative behaviors of the brokers are plotted for the case of different number of brokers. Unsurprisingly, the average price monotonically increases with the transmission delay, since larger transmission delay results in lower waiting delay and hence in a lower task rate acceptable by the public cloud. As plotted in Fig. 8b, the corresponding price reduction varies between 2 and 3 percent in the case of 10 brokers and 7.5 and 8.5 percent in the case of two brokers. That is to say, the price reduction ratio is only slightly affected as the transmission delay changes. This is because, while increasing the transmission delay enforces the brokers to outsource a larger portion of tasks to the on-demand requested cloud, the task rate acceptable by the on-demand cloud also decreases. From this experiment we conclude that cooperating gain may exist in a diverse set of networking settings with respect to transmission delay, and the cooperating gain is less affected by the transmission delay than the number of brokers.

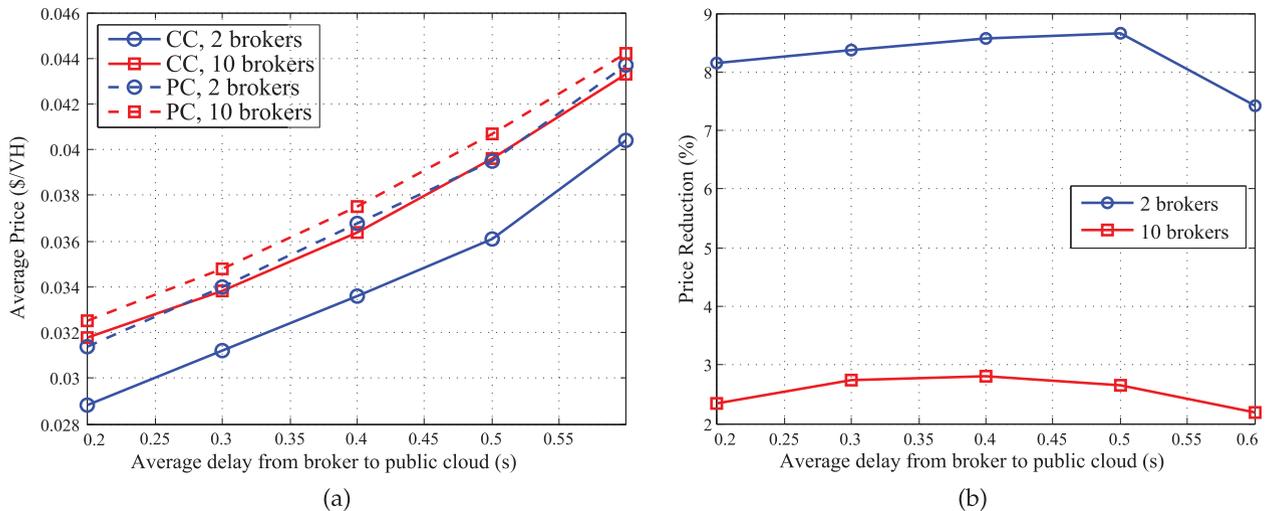


Fig. 8. (a) Average price and (b) price reduction with varying transmission delay between each broker and the public cloud for different number of brokers.

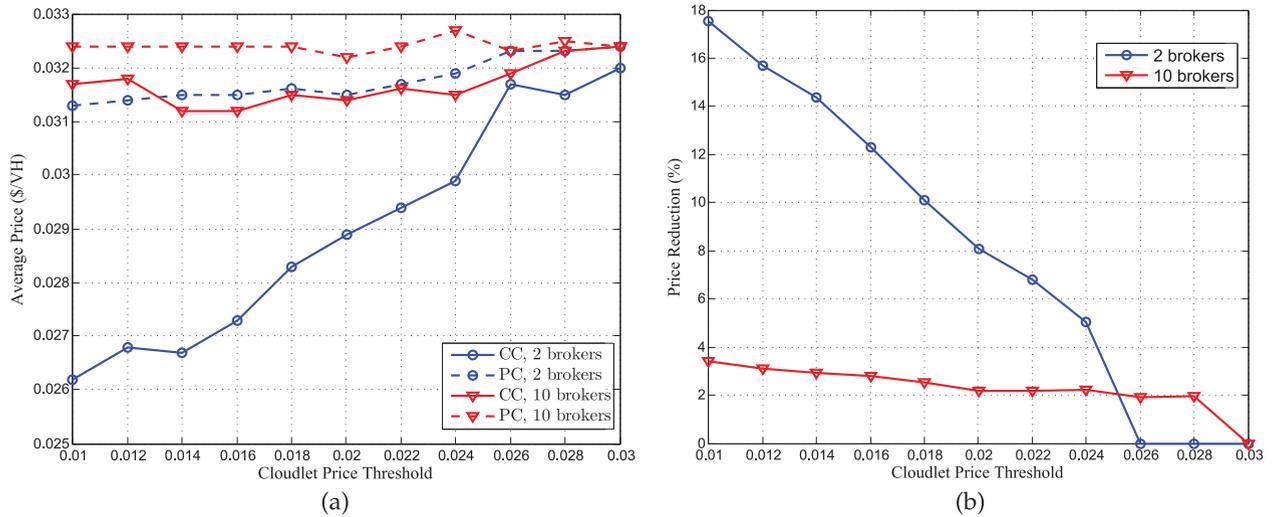


Fig. 9. (a) Average price and (b) price reduction with varying cloudlet threshold price for total offered task rate  $A_0$  between [50 100].

Finally, we study the effects of the price threshold of the cloudlet on the competitive and cooperative behaviors. In Fig. 9a, we plot the average price achievable by the PC and CC schemes, with 2 and 10 brokers, in the case of total offered task rate  $A_0$  between [50 100]. In the two-broker case, we find that the average price at the disagreement point varies between [0.031 0.032] and is only slightly affected by the threshold price, while it increases linearly from 0.026 to 0.032 in the cooperative scheme. As shown in Fig. 9b, the cooperation gain degrades from more than 17 percent to zero as the threshold price increases from 0.01 to 0.03. In the case of 10 brokers, a nearly constant price reduction around 3 percent can be achieved in a wide threshold price range. The corresponding results are plotted in Fig. 10 for the case of higher total offered task rate  $A_0$  between [100 150]. We find that cooperation gain can be achieved in a wider range of threshold price compared with in Fig. 9. For example, the price reductions achievable by PC and CC are 1.5 and 2.6 percent at cloudlet price threshold 0.03 in Fig. 10, while almost no price reduction can be achieved by PC or CC in Fig. 9. This is because the brokers have stronger incentive to outsource more computing tasks

to the low-delay cloudlet in the case of higher offered traffic load level, which results in higher bidding price at disagreement point in Fig. 10 compared to in Fig. 9a and hence possibly higher price reduction through cooperation. This experiment indicates that, there is no need for brokers to cooperate if the cloudlet threshold price becomes high and there is only a low number of brokers, and in the case of high cloudlet threshold price it is more desirable to cooperate with more demands of computing resources, i.e., more brokers and higher offered traffic load level.

## 7 CONCLUSIONS

We studied the problem of user cost minimization in MCC networks by considering a multi-broker scenario with heterogeneity of the cloud architecture and of the pricing strategies. We first studied a scenario in which brokers purely compete, and demonstrated theoretically the existence of disagreement points and convergence of the best response strategies of individual brokers to disagreement points. We then formulated a cooperative problem to minimize the average price achievable by all brokers so that

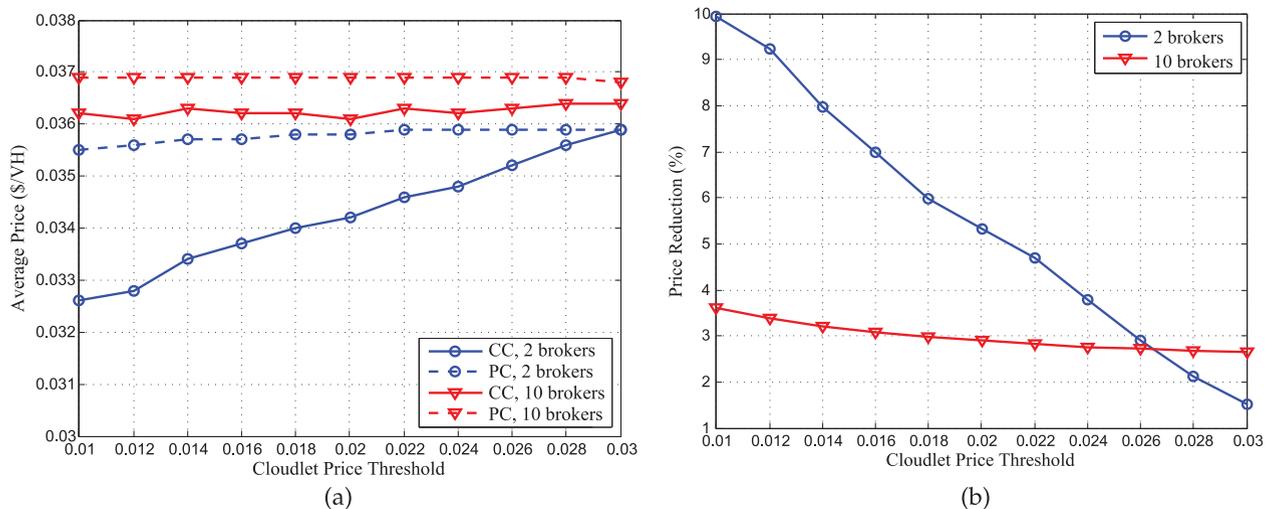


Fig. 10. (a) Average price and (b) price reduction with varying cloudlet threshold price for total offered task rate  $A_0$  between [100 150].

no broker has a price higher than the disagreement point. We obtained the globally optimal solution of the resulting non-convex problem through a newly designed algorithm. Through simulation results, we showed that considerable cooperative gains over pure competition can be achieved in markets with a few brokers only, and we highlighted the fact that the cooperative gain is low when there is a high number of brokers, i.e., the value of cooperation is marginal in this case.

**ACKNOWLEDGMENTS**

This work was supported in part by the US National Science Foundation under Grant CNS-1218717 and by the Air Force Research Laboratory under Contract FA8750-14-1-0074.

**REFERENCES**

[1] K. Kumar and Y.-H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" *IEEE Comput.*, vol. 43, no. 4, pp. 51–56, Apr. 2010.

[2] P. Bahl, R. Y. Han, L. E. Li, and M. Satyanarayanan, "Advancing the state of mobile cloud computing," in *Proc. ACM Int. Workshop Mobile Cloud Comput. Services*, Low Wood Bay, Lake District, U.K., Jun. 2012, pp. 21–28.

[3] H. P. Martinez, "Learning deep physiological models of affect," *IEEE Comput. Intell. Mag.*, vol. 8, no. 2, pp. 20–33, May 2013.

[4] S. Wang and S. Dey, "Adaptive mobile cloud computing to enable rich mobile multimedia applications," *IEEE Trans. Multimedia*, vol. 15, no. 4, pp. 870–883, Jun. 2013.

[5] Amazon EC2 [Online]. Available: <http://aws.amazon.com/ec2/> 2011.

[6] M. Satyanarayanan, P. Bahl, and R. Cáceres, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Comput.*, vol. 8, no. 4, pp. 14–23, Oct. 2009.

[7] Y. Wu, C. Wu, B. Li, X. Qiu, and F. C. Lau, "CloudMedia: When cloud on demand meets video on demand," in *Proc. IEEE Int. Conf. Distrib. Comput. Syst.*, Minneapolis, MN, USA, Jun. 2011, pp. 268–277.

[8] D. Niu, C. Feng, and B. Li, "A theory of cloud bandwidth pricing for Video-on-demand providers," in *Proc. IEEE Int. Conf. Comput. Commun.*, Orlando, FL, USA, Mar. 2012, pp. 711–719.

[9] M. R. Rahimi, N. Venkatasubramanian, S. Mehrotra, and A. Vasilakos, "MAPCloud: Mobile applications on an elastic and scalable 2-tier cloud architecture," in *Proc. IEEE/ACM Int. Conf. Utility Cloud Comput.*, Chicago, IL, USA, Nov. 2012, pp. 83–90.

[10] M. Feldman, K. Lai, and L. Zhang, "The proportional-share allocation market for computational resources," *IEEE Trans. Parallel Distrib. Syst.*, vol. 20, no. 8, pp. 1075–1088, Aug. 2009.

[11] M. Zafer, Y. Song, and K.-W. Lee, "Optimal bids for spot VMS in a cloud for deadline constrained jobs," in *Proc. IEEE Int. Conf. Cloud Comput.*, Honolulu, HI, USA, Jun. 2012, pp. 75–82.

[12] D. T. Hoang, D. Niyato, and P. Wang, "Optimal admission control policy for mobile cloud computing hotspot with cloudlet," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Paris, France, Apr. 2012, pp. 3145–3149.

[13] H. Liang, L. X. Cai, D. Huang, X. S. Shen, and D. Peng, "An SMDP-based service model for interdomain resource allocation in mobile cloud networks," *IEEE Trans. Veh. Technol.*, vol. 61, no. 5, pp. 2222–2232, Jun. 2012.

[14] D. Niyato, K. Zhu, and P. Wang, "Cooperative virtual machine management for multi-organization cloud computing environment," in *Proc. ACM Int. Workshop Game Theory Commun. Netw.*, Paris, France, May 2011, pp. 528–537.

[15] S. Di and C.-L. Wang, "Error-tolerant resource allocation and payment minimization for cloud system," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 6, pp. 1097–1106, Jun. 2013.

[16] S. Chaisiri, R. Kaewpuang, B.-S. Lee, and D. Niyato, "Cost minimization for provisioning virtual servers in amazon elastic compute cloud," in *Proc. IEEE Int. Symp. Model., Anal., Simul. Comput. Telecommun. Syst.*, Jul. 2011, pp. 85–95.

[17] W. Wang, B. Li, and B. Liang, "Towards optimal capacity segmentation with hybrid cloud pricing," in *Proc. IEEE Int. Conf. Distrib. Comput. Syst.*, Macau, China, Jun. 2012, pp. 425–434.

[18] F. Teng and F. Magoulès, "Resource pricing and equilibrium allocation policy in cloud computing," in *Proc. IEEE Int. Conf. Comput. Inf. Technol.*, Bradford, U.K., Jun. 2010, pp. 195–202.

[19] C.-W. Tsai and Z. Tsai, "Bid-proportional auction for resource allocation in Capacity-constrained clouds," in *Proc. IEEE Int. Conf. Adv. Netw. Appl. Workshops*, Fukuoka, Japan, Mar. 2012, pp. 1178–1183.

[20] S. Di and C.-L. Wang, "Dynamic optimization of multiattribute resource allocation in self-organizing clouds," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 3, pp. 464–478, Mar. 2013.

[21] G. Hardin, "The tragedy of the commons," *Science (AAAS)*, vol. 162, no. 3859, pp. 1243–1248, Dec. 1968.

[22] D. Bruneo, "A stochastic model to investigate data center performance and QOS in IaaS cloud computing systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 3, pp. 560–569, Mar. 2014.

[23] D. Bertsekas and R. Gallager, *Data Networks*. Englewood Cliffs, NJ, USA: Prentice-Hall, 2000.

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

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

[26] G. Scutari, D. P. Palomar, and S. Barbarossa, "Asynchronous iterative water-filling for gaussian frequency-selective interference channels," *IEEE Trans. Inf. Theory*, vol. 54, no. 7, pp. 2868–2878, Jul. 2008.

[27] H. D. Sherali and W. P. Adams, *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems*. Boston, MA, USA: Kluwer, 1999.



**Zhangyu Guan (M'11)** received the PhD degree in communication and information systems from Shandong University, Jinan, China, in 2010. He is currently a postdoctoral research associate with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA. He was a visiting PhD student with the Department of Electrical Engineering, The State University of New York (SUNY) at Buffalo, Buffalo, NY, from 2009 to 2010. He was a lecturer with Shandong University from 2011 to 2014, and he was a postdoctoral research associate with the Department of Electrical Engineering, SUNY Buffalo from November 2012 to August 2015. His current research interests are in cognitive radio and software-defined networking, wireless multimedia sensor networks, and underwater networks. He has served as a TPC member for IEEE INFOCOM 2016, IEEE GLOBECOM 2015, IEEE ICNC 2012-2015, and IEEE VTC 2011-Fall and 2015-Fall, among others, and served as a reviewer for the *IEEE Transactions on Communications*, the *IEEE Journal on Selected Areas in Communications*, and the *IEEE Communications Surveys and Tutorials*, among others. He is a member of the IEEE.



**Tommaso Melodia (M'07)** received the PhD degree in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, GA, in 2007. He is an associate professor with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA. His research has been supported by the US National Science Foundation, Air Force Research Laboratory, and the Office of Naval Research, among others. His current research interests are in modeling, optimization, and experimental evaluation of networked communication systems, with applications to ultrasonic intrabody networks, cognitive and cooperative networks, multimedia sensor networks, and underwater networks. He received the US National Science Foundation CAREER Award and coauthored a paper that was recognized as the ISI Fast Breaking Paper in the field of computer science for February 2009 and of an ACM WUWNet 2013 Best Paper Award. He was the Technical Program Committee vice chair for IEEE Globecom 2013 and the Technical Program Committee vice chair for information systems for IEEE INFOCOM 2013, and is an Area Chair for IEEE INFOCOM 2016. He serves on the editorial boards of the *IEEE Transactions on Mobile Computing*, the *IEEE Transactions on Wireless Communications*, the *IEEE Transactions on Multimedia*, and *Computer Networks*. He is a member of the IEEE.

▷ For more information on this or any other computing topic, please visit our Digital Library at [www.computer.org/publications/dlib](http://www.computer.org/publications/dlib).