# An Optimal Real-Time Distributed Algorithm for Utility Maximization of Mobile Ad Hoc Cloud

Fenghui Zhang, Ruilong Deng, *Member, IEEE*, and Hao Liang, *Member, IEEE*

***Abstract*— In this letter, we investigate utility maximization of mobile ad hoc cloud with an incentive mechanism to encourage mobile devices to share their idle resources. Considering that at different time slots the amount of resources demanded by the resource buyer (RB) is different and the revenue of per unit resource obtained by resource providers (RPs) is different, a real-time distributed algorithm is developed. First, by analyzing the preferences of the RB and RPs, the utility function and cost function are developed for them, respectively. Then, we propose a real-time distributed algorithm to find the maximum utility of the overall system under the price incentive mechanism, where the obtained optimal pricing can align the individual optimality with the overall system optimality. Simulation results confirm that the proposed algorithm can maximize the utility of the overall system compared with the state-of-the-art schemes.**

***Index Terms*— Convex optimization, mobile ad hoc cloud, offloading, utility maximization.**

## I. Introduction

As MOBILE devices are gaining enormous popularity, more and more mobile applications demanding intensive computation and high energy consumption are emerging and attracting great attentions [1]. However, mobile devices are in general resource-constrained, having limited computational resources and battery life. An efficient way to alleviate the resource scarcity of mobile devices is to offload their resource-demanding tasks to remote clouds [2]. However, in some places, wireless connections are weak, such that offloading tasks to remote clouds may cause huge inconvenience to customers. This problem can be solved by the mobile ad hoc cloud (MAC), which is composed of nearby mobile devices sharing their idle resources using the ad hoc network [3]. In the MAC, resource providers (RPs) afford their idle resources to satisfy the requirement of resource buyer (RB). The advantages are, firstly, the tasks can be done at a much lower price by mobile devices nearby [4]. Secondly, offloading tasks to mobile devices nearby through WLAN/WiFi can significantly reduce the communication latency [5], [6]. Thirdly, using the MAC can improve the communication efficiency and reduce the communication bandwidth usage [7].

Manuscript received December 16, 2017; revised January 17, 2018; accepted February 5, 2018. Date of publication February 12, 2018; date of current version April 7, 2018. This work was supported in part by the Universities Excellent Young Talents Foundation of Anhui Province, China (No. gxfxZD2016199), the Alberta Innovates-Technology Futures post-doctoral fellowship, and a research grant from the Natural Sciences and Engineering Research Council of Canada. The associate editor coordinating the review of this paper and approving it for publication was R. Aguero.
*(Corresponding author: Ruilong Deng.)*

F. Zhang is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 1H9, Canada, and also with the School of Electronics and Information Engineering, West Anhui University, Lu'an 237012, China (e-mail: zhangfenghui@wxc.edu.cn).

R. Deng and H. Liang are with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 1H9, Canada (e-mail: ruilong@ualberta.ca; hao2@ualberta.ca).

Digital Object Identifier 10.1109/LCOMM.2018.2804928

An important aspect of research on the MAC is the incentive mechanism [8]. In [6], a bill backlog on each mobile device is raised to encourage mobile devices to share their resources. If a device's bill backlog is larger than a threshold, it will be unable to get more services from others. In [9], a directory-based framework is proposed to keep track of the retribution and reward valuations for devices even after they move from one ad hoc environment to another. These mechanisms enforce mobile devices in the MAC to share their resources. However, it is more reasonable to use the strategy of encouragement. In [5], an incentive mechanism in the mobile cloud has been studied. However, it mainly focuses on the supply side of mobile cloud. In [4], a Stackelberg game approach is used to improve the initiative of mobile devices. The service buyer decides the price while the service providers decide the amount of resources that will be provided to the buyer. In [10], a double-sided bidding mechanism for resource sharing is used in the MAC. In these two studies, although both supply and demand sides are considered, they never consider the utility of overall system. In [11], an incentive-based workload assignment problem is formulated to maximize the overall system utility and is solved by a centralized scheme. Besides, an incentive decentralized scheme is proposed based on game theoretic approach to model the interaction among mobile devices and maximize the individual utility. However, in aforementioned literatures, the relationship between maximizing the overall system utility and maximizing the individual utility has never been thoroughly investigated. To this end, we propose to optimize the utility of MAC by using a pricing mechanism to encourage RPs to share their idle resources while satisfying the resource requirement of RB. The pricing mechanism designed in this letter is to seek an appropriate pricing which can align the individual optimality with the overall system optimality. Under this pricing, when each selfish individual tries to maximize its own utility, they are equivalently achieving the global optimization. Specifically, the original contributions are threefold:

1) We analyze the RPs' and RB's preferences and model their cost and utility patterns in form of carefully selected functions based on concepts from microeconomics;
2) A real-time distributed algorithm is proposed for the MAC to maximize the utility of overall system by encouraging RPs to share their idle resources to the RB, and the obtained optimal pricing can align the individual optimality with the overall system optimality;
3) The proposed algorithm is evaluated and compared with existing schemes, and simulation results confirm that the system will benefit from the proposed algorithm.

## II. System Model

A MAC consist of $N+1$ mobile devices is considered, where one RB uses resources from other devices, while the other $N$ mobile devices are RPs. The process of resource requirement and provision in the MAC can be divided into a set $K \triangleq \{1,\ldots,k,\ldots\}$ of time slots. The amount of resources required by the RB has an upper bound, denoted by $y_{\max}$. We use $y_k$ to denote the amount of resource requirement at time slot $k$, so we have $0 \le y_k \le y_{\max}$. The amount of resources provided by the $i^{\text{th}}$ RP at time slot $k$ is represented by $x_k^i$, ($i \in I$, where $I$ is the set of all RPs). We use $x_{\max}$ to denote the maximum amount of resources provided by one RP, so we have $0 \le x_k^i \le x_{\max}$. The practical scenarios of the system model include the high-frame-rate video streaming [2], context sensitive offloading [3], cooperative task execution [4], and so on. These kinds of applications can be run on the MAC and require different computing resources in different time slots.

### A. Cost Function of RP

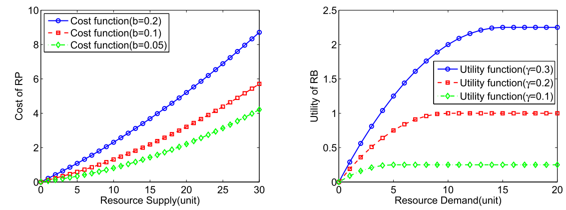
The different responses of different RPs to offering resources can be modelled analytically by adopting the concept of cost functions from microeconomics [12], [13]. For each RP, the cost function is represented as $C(x,b)$, where $x$ is the level of resources provided by the RP and $b$ denotes other cost factors, e.g., the workload, the battery power that has been used, etc. Specifically, we assume that the cost functions of RPs fulfill the following properties:

1) Property I: Cost functions are non-decreasing. Mathematically, this implies $\frac{\partial C(x,b)}{\partial x} \ge 0$.
2) Property II: The marginal cost of RPs is a non-decreasing function, so we have $\frac{\partial^2\ C(x,b)}{\partial x^2} \ge 0$.
3) Property III: We are able to rank the RPs based on their costs. In our formulation, we assume that for a fixed afford level $x$, a larger $b$ implies a higher cost, which can be expressed as $\frac{\partial C(x,b)}{\partial b} \ge 0$.
4) Property IV: When the RP doesn't provide any resource, it's cost function value is equal to zero, i.e., $C(0,b)=0$.
5) Property V: When affording resources, each RP has its own bearable point such that under this point the cost is only due to the battery power consumption (i.e., the cost function is linear). When the afford amount is larger than this point, it will not only consume the battery power but also affect the program running on the RP, and thus the cost will raise rapidly.

In this letter, we consider the quadratic cost functions. The bearable point is the tangency of each curve. That is, $C(x,b) = \begin{cases} ax^2+bx+c, & \text{if } x > \sqrt{\frac{c}{a}} \\ (2\sqrt{ac}+b)x, & \text{otherwise} \end{cases}$, where $a$, $c$ are pre-determined parameters. Sample cost functions are shown in Fig. 1(a).

### B. Utility Function of RB

The resource amount required by the RB is changeable since the RB may have different computing tasks at different time slots. Like in [10], we define resources as the CPU cycles, and they are continuous in the domain. The revenue of RB can be modelled by adopting the concept of utility functions from microeconomics. We use $U(y,\gamma)$ to denote the utility of RB,




(a) Cost function of RP (a=0.003, c=0.01). (b) Utility function of RB ($\alpha$=0.01).

Fig. 1. Sample functions of the RPs and RB in the MAC.

in which $y$ represents the amount of resource requirement, and $\gamma$ represents different states of the RB. For instance, the RB requires more resources if the computing task is larger. Specifically, we assume that the utility function of RB has the following properties:

1) Property I: Utility functions are non-decreasing, so we have $\frac{\partial U(y,\gamma)}{\partial y} \ge 0$.
2) Property II: The marginal benefit of the RB is a non-increasing function. Thus, the level of satisfaction for the RB will gradually get saturated, so we have $\frac{\partial^2\ U(y,\gamma)}{\partial y^2} \le 0$.
3) Property III: We rank different resource requirements based on their utilities. In our formulation, a larger $\gamma$ implies a higher utility, so we have $\frac{\partial U(y,\gamma)}{\partial \gamma} \ge 0$.
4) Property IV: When there is no resource requirement, the utility function value is equal to zero, i.e., $U(0,\gamma)=0$.
5) Property V: When the RB has required a certain amount of resources, the utility obtained from demanding more resources will no longer increase, since its own resource can deal with the remaining task easily.

In this letter, we consider the quadratic utility function. That is, $U(y,\gamma) = \begin{cases} \gamma y - \alpha y^2, & \text{if } 0 \le y \le \frac{\gamma}{2\alpha} \\ \frac{\gamma^2}{4\alpha}, & \text{otherwise} \end{cases}$, where $\alpha$ is a pre-determined parameter. Sample utility functions are shown in Fig. 1(b).

## III. Problem Formulation and Solution

In this system, we use a pricing mechanism to encourage RPs to share their idle resources. Within each time slot, the RB announces the price of per unit resource, and RPs provide resources according to the price. Note that in the MAC, the objective of each RP is to maximize the received revenue from selling resources minus the cost function, while the objective of the RB is to maximize the utility function minus the paid expense in buying resources. In practice, from the system perspective, it is desirable that the utility of RB is maximized and the sum of cost of all RPs is minimized. Thus, the utility of system can be represented by the utility function of RB minus the sum of cost functions of all RPs. In the following, we combine RPs and RB into one model in order to investigate the relationship between maximizing the overall system utility and maximizing the individual utility.

### A. Problem Formulation

To maximize the utility of MAC, it is desirable to use the utility of RB minus all the cost of RPs in the system as the optimization objective, while keeping the total supply level of all RPs satisfying the resource requirement of RB.

An efficient resource requirement process can be characterized as the solution to the following problem:

$$\begin{aligned} \max_{\substack{0\le x_k^i \le x_{\max} \\ 0 \le y_k \le y_{\max}}} \quad & \sum_{k\in K} U(y_k,\gamma) - \sum_{k\in K}\sum_{i\in I} C_i(x_k^i, b) \\ \text{s.t.}\ & y_k \le \sum_{i\in I} x_k^i \quad \forall k \in K. \end{aligned} \tag{1}$$

Note that the formulation (1) is a time-slotted model, which can be solved independently at each time slot, so we get the following formulation at a certain time slot $k \in K$:

$$\begin{aligned} \max_{\substack{0\le x_k^i \le x_{\max} \\ 0 \le y_k \le y_{\max}}} \quad & U(y_k,\gamma) - \sum_{i\in I} C_i(x_k^i, b) \\ \text{s.t.}\ & y_k \le \sum_{i\in I} x_k^i. \end{aligned} \tag{2}$$

### *B. Dual Decomposition Approach*

Since the formulation (2) is concave, which can be solved by Lagrange duality, so we have the following Lagrangian dual function:

$$L(\boldsymbol{x}_k, y_k, \lambda^k) = U(y_k,\gamma) - \sum_{i\in I} C_i(x_k^i, b) - \lambda^k \ (y_k - \sum_{i\in I} x_k^i),$$

where $\lambda^k$ is the Lagrange multiplier in time slot $k$, and $\boldsymbol{x}_k \triangleq [x_k^1, \ldots, x_k^i, \ldots]$. We can write the objective function of the dual optimization problem as

$$q(\lambda^k) = \max_{\substack{0\le x_k^i \le x_{\max} \\ 0 \le y_k \le y_{\max}}} L(\boldsymbol{x}_k, y_k, \lambda^k) = \sum_{i\in I} P_k^i(\lambda^k) + B_k(\lambda^k),$$

where

$$P_k^i(\lambda^k) \triangleq \max_{0\le x_k^i \le x_{\max}} \lambda^k x_k^i - C_i(x_k^i, b) \tag{3}$$

$$B_k(\lambda^k) \triangleq \max_{0 \le y_k \le y_{\max}} U(y_k,\gamma) - \lambda^k y_k. \tag{4}$$

The formulation $q(\lambda^k)$ denotes the utility of system. From the actual system, we can conclude that $\lambda^k$ denotes the price that the RB announces, so the formulation (3) denotes the maximum utility of all RPs at time slot $k$ with the amount of resources provided at the price $\lambda^k$. The formulation (4) denotes the maximum utility of RB at time slot $k$ with the amount of resources demanded at the price $\lambda^k$. Then, the dual problem is

$$\min_{\lambda^k > 0} \ q(\lambda^k). \tag{5}$$

In this system, the strong duality holds, so we can solve the dual problem (5) instead of the primal problem (2). In this case, we can obtain the optimal solution $\lambda^{k*}$ of the dual problem. Then, each individual RP and RB can simply solve their own local optimization problem determined by (3) and (4) to obtain $x_k^{i*}$ and $y_k^*$, respectively.

*Remark 1*: The objective function (2) is to maximize the overall system utility, while the objective functions (3) and (4) are to maximize the individual utility. For an arbitrary price $\lambda^k$, the locally optimal solution $x_k^i$ and $y_k$ to (3) and (4) may not be globally optimal solution to (2). However, by duality theory [14], there exists a dual optimal price $\lambda^{k*}$ such that the locally optimal solution $x_k^{i*}$ and $y_k^*$ will be globally optimal. That is, the pricing mechanism designed in this letter is to seek the appropriate pricing which can align the individual optimality with the overall system optimality. The following distributed algorithm is proposed to obtain the optimal price $\lambda^{k*}$ in a distributed manner. Then, under the appropriate pricing, when each selfish individual tries to maximize its own utility, they are equivalently achieving the global optimization.

### *C. Distributed Algorithm*

From the Lagrange duality theory, we know that by giving the system with the solution $\lambda^{k*}$ of the dual problem, we can achieve the solution of the primal problem (2). Interestingly, it is possible to solve the dual problem in an iterative manner using the gradient projection method,

$$\lambda_{\tau+1}^k = [\lambda_\tau^k - \mu \frac{\partial q(\lambda_\tau^k)}{\partial \lambda_\tau^k}] = \{\lambda_\tau^k - \mu[\sum_{i\in I} x_k^{i*} - y_k^*]\}, \tag{6}$$

where $\tau$ is the iteration index, $\mu$ is the step size of iterations, and $x_k^{i*}$, $y_k^*$ are the local optimizers of (3), (4), respectively.

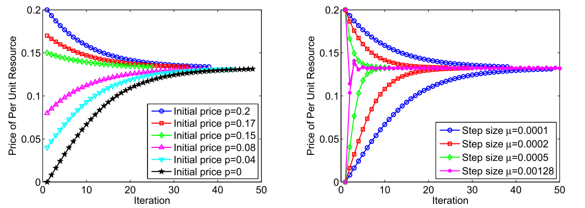
**Algorithm 1** Executed by the RB at Time Slot $k \in K$

**Input**: the initial price $\lambda_\tau^k$
**repeat**
  Broadcast the new value of $\lambda_\tau^k$ to all RPs;
  Update the resource requirement $y^k\ (\lambda_\tau^k)$ using (4);
  Wait until receive all the optimal amount of provided resources $x_k^{i}(\lambda_\tau^k)$ from RPs;
  Update the total amount $\sum_{i\in I} x_k^{i*}(\lambda_\tau^k)$;
  Compute the new value of $\lambda_{\tau+1}^k$ using (6);
**until** $|\lambda_\tau^k - \lambda_{\tau+1}^k| \le \varepsilon$;
**Output**: the optimal price $\lambda^{k*}$

**Algorithm 2** Executed by Each RP at Time Slot $k \in K$

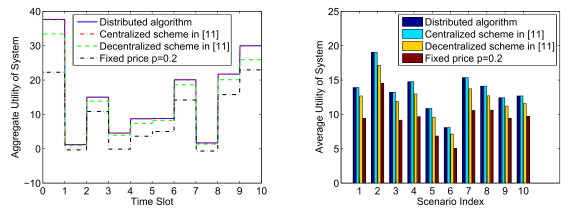
**Input**: the price $\lambda_\tau^k$
1 Receive the new value of $\lambda_\tau^k$ from the RB;
2 Update the optimal amount of provided resources $x_k^{i*}(\lambda_\tau^k)$ by each RP using (3);
3 Send the updated $x_k^{i*}(\lambda_\tau^k)$ to the RB;
**Output**: the optimal amount of resources $x_k^{i*}(\lambda_\tau^k)$

We propose a distributed algorithm to solve this problem. In each time slot, the algorithms for the RB and RPs are summarized in **Algorithm 1** and **Algorithm 2**, respectively. In **Algorithm 1**, the RB starts with the price $\lambda^k$ and announces it to all RPs, then the RB calculates the amount of resources demanded at the price $\lambda^k$ using (4). When the RB receives all values of resources provided by RPs, it uses (6) to compute a new price. The loop continues until the price converges, and then the system takes this price as the optimal price. Considering **Algorithm 2**, in line 1, each RP starts with its initial condition, where the parameter is randomly determined from the domain. Lines 2 to 4 describe the response of each RP to the newly announced price $\lambda^k$. Each RP receives the new value of $\lambda^k$ in line 3 and solves the local problem (3) to get the optimal provided resources $x_k^{i*}(\lambda^k)$. Then, all RPs communicate the new value of $x_k^{i*}(\lambda^k)$ to the RB.




(a) Convergence of price using different initial prices. (b) Convergence of price using different step sizes.

Fig. 2. The convergence of our proposed algorithm.




(a) The comparison of aggregate utility of system for 10 time slots. (b) The comparison of system average utility after running 100 times.

Fig. 3. The comparison of system utility for different algorithms.

## IV. Performance Evaluation

In this section, we present simulation results and assess the performance of our proposed algorithm. In the simulation model, there are a total of 5 RPs. The process of the resource requirement by RB is divided into 10 time slots. The time slot duration is one hour [4]. In each time slot, the idle resources of different RPs are different, so in the cost functions of RPs, all the parameters $b$ are selected randomly from the interval [0.03, 0.7] and remain fixed within one time slot. As depict in [15], since virtual machines of Amazon have different prices, we use the upper and lower bounds of the price as the average bound cost of RPs. Accordingly, we set $a = 0.001$ and $c = 0.01$. The resource requirements of RB are different in different time slots, so in the utility function of RB, all the parameters $\gamma$ are selected randomly from the interval [0, 1] and remain fixed within one time slot. We set $\alpha = 0.005$ and the convergence condition is $\varepsilon = 10^{-4}$.

The convergence of algorithm for one time slot is shown in Fig. 2. As illustrated in Fig. 2(a), we can conclude that from any initial price of per unit resource, it finally converges to the optimal price using the distributed algorithm. Fig. 2(b) shows the convergence of algorithm using different step sizes. When the step size is larger, the convergence becomes faster; but when the step size is larger than a certain range, the system starts to vibrate. We can observe from the simulation results that when $\mu > 0.00128$ the system will no longer converge. Note that if the initial price and step size are chosen more reasonably, the convergence would be even faster. The average computational complexity per iteration is around microseconds. From the above, the complexity of algorithm is very low, which can accommodate the real-world applications of mobile devices with favorable scalability.

For comparison with our proposed algorithm, we consider the centralized and decentralized schemes in [11], and the fixed price as a benchmark. In Fig. 3(a), we compare the system utility of different algorithms for 10 time slots. Fig. 3(b) shows that the system runs 100 times using random parameters from the domain. We can conclude that our proposed algorithm and the schemes in [11], by considering the interaction between RPs and the RB, can achieve higher utility than that of the fixed price. On the other hand, since the decentralized scheme in [11] is based on game theoretic approach to maximize the individual utility, the aggregate utility of system is not the best. In our distributed algorithm, the obtained optimal pricing can align the individual optimality with the overall system optimality, which is the novelty of our letter.

## V. Conclusion

In this letter, we propose an optimal real-time distributed algorithm in the MAC based on utility maximization. It can be implemented in a distributed manner to maximize the aggregated utility of RB and RPs while encouraging RPs to share their idle resources. The obtained optimal pricing can align the individual optimality with the overall system optimality. Simulation results confirm that by using our proposed optimization-based real-time distributed algorithm, not only the system could benefit, but also the resources provided by RPs will satisfy the requirement of RB. The idea developed in this letter can be further extended in several directions. In particular, the system with multiple RBs can be considered, and the impact of malicious RPs can also be explored.

## References

[1] T. Soyata, R. Muraleedharan, C. Funai, M. Kwon, and W. Heinzelman, “Cloud-vision: Real-time face recognition using a mobile-cloudlet-cloud acceleration architecture,” in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jul. 2012, pp. 59–66.
[2] J. Wu, C. Yuen, N.-M. Cheung, J. Chen, and C. W. Chen, “Enabling adaptive high-frame-rate video streaming in mobile cloud gaming applications,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 12, pp. 1988–2001, Dec. 2015.
[3] B. Zhou, A. V. Dastjerdi, R. N. Calheiros, S. N. Srirama, and R. Buyya, “A context sensitive offloading scheme for mobile cloud computing service,” in *Proc. IEEE Int. Conf. Cloud Comput. (CLOUD)*, Jun./Jul. 2015, pp. 869–876.
[4] X. Wang, X. Chen, W. Wu, N. An, and L. Wang, “Cooperative application execution in mobile cloud computing: A Stackelberg game approach,” *IEEE Commun. Lett.*, vol. 20, no. 5, pp. 946–949, May 2016.
[5] S. Al Noor, R. Hasan, and M. M. Haque, “CellCloud: A novel cost effective formation of mobile cloud based on bidding incentives,” in *Proc. IEEE Int. Conf. Cloud Comput. (CLOUD)*, Jun./Jul. 2014, pp. 200–207.
[6] J. Song, Y. Cui, M. Li, J. Qiu, and R. Buyya, “Energy-traffic tradeoff cooperative offloading for mobile cloud computing,” in *Proc. IEEE Int. Symp. Quality Service (IWQoS)*, May 2014, pp. 284–289.
[7] AT&T Inc. (2011). *Annual Report 2011 Getting to the Future First*. [Online]. Available: https://www.att.com/Common/about_us/annualreport/pdfs/att_ar_2011_highlights.pdf
[8] I. Yaqoob, E. Ahmed, A. Gani, S. Mokhtar, M. Imran, and S. Guizani, “Mobile ad hoc cloud: A survey,” *Wireless Commun. Mobile Comput.*, vol. 16, no. 16, pp. 2572–2589, 2016.
[9] A. Yousafzai, V. Chang, A. Gani, and R. M. Noor, “Directory-based incentive management services for ad-hoc mobile clouds,” *Int. J. Inf. Manage.*, vol. 36, no. 6, pp. 900–906, 2016.
[10] L. Tang, S. He, and Q. Li, “Double-sided bidding mechanism for resource sharing in mobile cloud,” *IEEE Trans. Veh. Technol.*, vol. 66, no. 2, pp. 1798–1809, Feb. 2017.
[11] B. Cao, S. Xia, Y. Li, and B. Li, “An incentive-based workload assignment with power allocation in ad hoc cloud,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.
[12] A. Mas-Colell, M. D. Whinston, and J. R. Green, *Microeconomic Theory*. London, U.K.: Oxford Univ. Press, 1995.
[13] M. Fahrioglu and F. L. Alvarado, “Using utility information to calibrate customer demand management behavior models,” *IEEE Trans. Power Syst.*, vol. 16, no. 2, pp. 317–322, May 2001.
[14] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[15] B. Javadi, R. K. Thulasiramy, and R. Buyya, “Statistical modeling of spot instance prices in public cloud environments,” in *Proc. IEEE Int. Conf. Utility Cloud Comput. (UCC)*, Dec. 2011, pp. 219–228.