This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2018.2847337, IEEE Transactions on Mobile Computing

Dynamic Computation Offloading for Mobile Cloud Computing: A Stochastic Game-Theoretic Approach

Jianchao Zheng, *Member, IEEE*, Yueming Cai, *Senior Member, IEEE*, Yuan Wu, *Senior Member, IEEE*, and Xuemin (Sherman) Shen, *Fellow, IEEE*

Abstract—Driven by the growing popularity of mobile applications, mobile cloud computing has been envisioned as a promising approach to enhance computation capability of mobile devices and reduce the energy consumptions. In this paper, we investigate the problem of multi-user computation offloading for mobile cloud computing under dynamic environment, wherein mobile users become active or inactive dynamically, and the wireless channels for mobile users to offload computation vary randomly. As mobile users are self-interested and selfish in offloading computation tasks to the mobile cloud, we formulate the mobile users' offloading decision process under dynamic environment as a stochastic game. We prove that the formulated stochastic game is equivalent to a weighted potential game which has at least one Nash Equilibrium (NE). We quantify the efficiency of the NE, and further propose a multi-agent stochastic learning algorithm to reach the NE with a guaranteed convergence rate (which is also analytically derived). Finally, we conduct simulations to validate the effectiveness of the proposed algorithm and evaluate its performance under dynamic environment.

Index Terms—Mobile cloud computing, multi-user computation offloading, dynamic environment, stochastic game, multi-agent stochastic learning.

1 Introduction

With the proliferation of smart wireless devices and mobile internet services, more and more mobile applications such as interactive gaming, face recognition, and augmented reality have emerged and drawn increasing interests [2]. These sophisticated applications usually require significant amounts of computation resources and energy consumptions, which, however, cannot be directly afforded by most mobile devices due to their limited computation resources and battery capacities [3], [4]. Therefore, mobile cloud computing, which enables mobile devices to offload their computation tasks to the resource-rich cloud infrastructures (such as Amazon EC2, Microsoft Azure, and Google App Engine) via wireless links, has been envisioned as a promising approach to address this challenge issue [5]. In cloud infrastructure, each mobile device is associated with a system-level cloud clone running a virtual machine that executes mobile applications on behalf of the mobile device [6].

Offloading mobile users' computation tasks to the mobile cloud infrastructure usually involves considerable communication burdens between the cloud and mobile devices, which thus necessitate a careful design of multi-user computation offloading strategy to improve wireless access efficiency [7]. A motivating example is as follows. When many mobile devices aggressively offload their computation tasks to a mobile cloud over the same wireless channel, they may generate severe co-channel interference to each other. Such a severe interference leads to lower offloading rates (i.e., mobile users' achievable data rates for sending computation tasks to the mobile cloud over the wireless link) and higher energy consumptions for mobile devices, which consequently compromise the benefit of offloading computation tasks. Therefore, it is very important to achieve an efficient computation offloading coordination when many mobile devices compete for a limited number of wireless channels to offload computation tasks to the mobile cloud infrastructure.

Game theory is a widely adopted mathematical tool to model and analyze complicated decision-making processes among a group of rational decision-makers of conflicting objectives [8], [9], [10], [11], [12]. Since different mobile devices are usually owned by different users, it is natural to adopt game theory to analyze the computation offloading process for multiple mobile users who exploit a common set of wireless channels to offload their computation tasks. Specifically, each user is modeled as a rational game player that observes and reacts to other users' offloading strategies in the *best response* manner. Such an interactive decision process is expected to reach an equilibrium point (also referred to as Nash Equilibrium, NE), at which no individual user will change its offloading strategy unilaterally. Moreover, by leveraging the intelligence of mobile users, game theory is useful for designing decentralized mechanisms with low complexity, which help to ease the heavy controlling and signaling overhead of complex centralized management [10].

However, applying game theory to model the multi-user computation offloading process should carefully deal with the complex real network environment. Specifically, mobile users may become active or inactive[1] dynamically, and wireless channels are also

- *This work is supported by the Jiangsu Provincial Natural Science Foundation of China under Grant BK20170755, by the National Postdoctoral Program for Innovative Talents of China under Grant BX201700109, by the Zhejiang Provincial Natural Science Foundation of China under Grant LR17F010002, and by the Natural Science and Engineering Research Council (NSERC), Canada. This paper has been presented in part at the IEEE/CIC ICCC Conference [1], July 2016, Chengdu, China.*
- *J. Zheng is with the National Innovation Institute of Defense Technology, Academy of Military Sciences PLA, Beijing, 100010, China. He is also with the College of Communications Engineering, PLA Army Engineering University, Nanjing, 210007, China (email: longxingren.zjc.s@163.com).*
- *Y. Cai is with the College of Communications Engineering, PLA Army Engineering University, Nanjing, 210007, China (email: caiym@vip.sina.com).*
- *Y. Wu is with the College of Information Engineering, Zhejiang University of Technology, Hangzhou, 310023, China (email: iewuy@zjut.edu.cn).*
- *X. Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (email: sshen@uwaterloo.ca).*

1. A mobile user is active if it has a computation task to be executed, while a mobile user is inactive (or silent) if it does not have a computation task to be executed.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2018.2847337, IEEE Transactions on Mobile Computing

time-varying. To capture the dynamics of network environment, in this paper, we adopt a novel stochastic game-theoretic approach to analyze the users' computation offloading decision-making process under dynamic conditions, and then propose a multi-agent stochastic learning algorithm to reach the NE of the stochastic game. The main contributions of this paper are summarized as follows:

- We formulate a stochastic game to model and analyze the multi-user computation offloading problem under dynamic environment, wherein both mobile users' activeness and wireless channel gains are time-varying. To show the existence of NE in the formulated stochastic game, we prove its equivalence to a weighted potential game which has at least one NE. Moreover, we analyze the performance bound of the NE in terms of the system cost and the number of mobile users who can benefit from cloud computing.
- To reach the NE of the formulated stochastic game, we propose a multi-agent stochastic learning algorithm for the multi-user computation offloading under dynamic environment. The proposed algorithm runs in a fully distributed manner without any information exchange, i.e., each user independently adjusts its offloading strategy based on its received action-reward instead of knowing other users' detailed offloading-strategies.
- As an important technical contribution in this paper, we theoretically derive the convergence rate of the multi-agent stochastic learning algorithm. It is technically challenging to prove the convergence property of the designed learning algorithms with multi-user interactions under dynamic environment, and our study here is the first one successfully addressing this issue.

The rest of this paper is organized as follows. In Section II, we give a brief review of the related works. In Section III, our system model is introduced. In Section IV, we propose a stochastic game to investigate the problem of dynamic computation offloading. In Section V, the performance of the NE of the game is analyzed. In Section VI, we propose a multi-agent stochastic learning algorithm to find the NE under dynamic environment. Section VII presents simulation results and discussions. Conclusions are drawn in Section VIII.

2 Related Work

In the literature, many existing works have studied the computation offloading problem from the perspective of a single mobile user. Rudenko *et al.* [17] used experimental results to show that computation offloading can save significant energy. In [18], the authors designed an adaptive timeout scheme for computation offloading to improve the energy savings on mobile devices. Wen *et al.* [6] proposed an optimization scheme for energy-efficient application execution on the cloud-assisted mobile application platform. Huertacanepa and Lee [19] proposed an adaptive application offloading mechanism based on both the current system conditions and the execution history of applications. By invoking the Lyapunov optimization, [20] and [21] studied the dynamic computation offloading policies for minimizing CPU and network energy consumption under real network environment. In [22], the authors modeled the unstable network as an alternating renewal process and proposed an offloading decision model for mobile cloud application.

Only a few works have discussed the computation offloading problem in the multi-user case. Yang *et al.* [23] proposed a genetic algorithm to solve the partition problem of wireless network

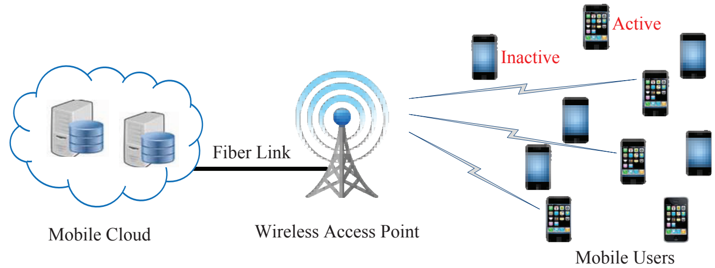


Fig. 1. An illustration of the system model (a group of mobile users offload computation tasks to a mobile cloud via a set of available wireless channels. Each mobile user could dynamically change its activeness, and the wireless channels are time-varying).

bandwidth among multiple users, which achieves high throughput of processing the streaming data. In [3], the authors devised a low-complexity heuristic method to perform energy-efficient task offloading for multiuser mobile cloud computing, while satisfying the delay requirements. Sardellitti *et al.* [4] proposed an iterative algorithm to perform the joint optimization of radio and computational resources for multi-cell mobile-edge computing, under latency and power budget constraints.

The above studies all belonged to the centralized computation offloading mechanisms which did not consider the interactions among multiple self-organizing users when they independently chose their computation offloading strategies. [10], [11], [12], [13], [14], [15], [16] modeled mobile users as self-interested game players and proposed decentralized mechanisms to solve the multi-user computation offloading problems. These previous studies mainly focused on the computation offloading problems under relatively static environment. However, in real network environment, due to dynamic mobile users' activeness and time-varying wireless channels, the utility of each player is dynamically varying, and thus the equilibrium solution of the static game model may never be reached. In this paper, we study the multi-user computation offloading problem with consideration of the dynamics of users' behaviors and time-varying channels, and adopt the theory of stochastic game that accounts for all the possible states of the dynamic process to successfully solve the problem.

The task of achieving NE solutions of the stochastic game in the distributed and dynamic environment is challenging. Most existing algorithms, such as the best (or better) response [24], fictitious play [25], spatial adaptive play [26], and no-regret learning [27] require enormous information exchanges for users' strategy updating and require the environment to be unchanged until reaching convergence of the algorithms. For the distributed and dynamic environment, some efficient algorithms have been proposed by invoking the stochastic learning automata (SLA) [28], [29], [30]. Specifically, the convergence of SLA-based algorithms to NE has been established for coordination games [28] and exact potential games [29], [30]. In this paper, we show the formulated stochastic game is equivalent to a weighted potential game, and then prove the convergence of the designed multi-agent stochastic learning algorithm to the NE of the weighted potential game. Moreover, to the best of our knowledge, our work is the first to establish the convergence rate of the SLA-based algorithm in the distributed and dynamic environment.

3 System Model

As shown in Figure 1, we consider a group of mobile users $\mathcal{N} = \{1, 2, \ldots, N\}$, where each user i may have a computationally intensive task $\mathcal{T}_i$ to be completed. There exists a wireless access

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2018.2847337, IEEE Transactions on Mobile Computing

TABLE 1
Summation of Used Notations

Notations	Description	Notations	Description
$\mathcal{N}$	set of mobile users	u_i^0	payoff function for static game $\mathcal{G}_{0,\Lambda}$
$\mathcal{M}$	set of wireless channels	u_i	payoff function for static game $\mathcal{G}_{1,\Lambda}$
s_i	user i's offloading strategy	$\bar{u}_i$	payoff function for stochastic game $\mathcal{G}_2$
θ_i	user i's active probability	$\mathcal{A}$	set of active users
p_i	user i's transmit power	$\boldsymbol{s}_{\mathcal{A}}$	strategy profile of all active users
$g_{i,o}$	instantaneous channel gain	$\boldsymbol{s}_{\mathcal{A}\backslash\{i\}}$	strategy profile of all active users excluding i
$\bar{g}_{i,o}$	expected channel gain	$\boldsymbol{s}$	strategy profile of all users
R_i	user i's data rate	$\boldsymbol{s}_{-i}$	strategy profile of all users excluding i
I_i	user i's received interference	$\mathbf{w}_i$	user i's strategy selection probability vector
V_i^{clo}	cloud computing cost	b	learning step-size of the proposed algorithm
V_i^{loc}	local computing cost	r_i^t	user i's action-reward at time t

point (AP) through which the mobile users can offload their computation tasks to the cloud center deployed by the telecom operator. Here, the wireless AP could be a WiFi access point, 3G/4G macro-cell or small-cell base station. Suppose that there are M available wireless channels denoted as $\mathcal{M} = \{1, 2, \ldots, M\}$. We use $s_i \in \{0\} \bigcup \mathcal{M}$ to denote mobile user i's computation offloading strategy. Specifically, $s_i > 0$ denotes that user i chooses to offload the computation task to the mobile cloud via wireless channel s_i; in opposite, $s_i = 0$ denotes that user i decides to compute its task locally without offloading to mobile cloud. Notice that for each user i, choosing different channels will lead to different offloading rates when it sends computation task to the AP, which consequently yields different costs.

3.1 Dynamics of Mobile Users' Activeness and Wireless Channels

Communication systems operate in a time-slotted fashion over time slots of equal duration (e.g., several microseconds or milliseconds [32]). A computation offloading period (e.g., several seconds [10]) usually consists of multiple time slots. In this paper, we consider a general and practical case that mobile users may become active or inactive dynamically within different time slots. Specifically, a mobile user is active if it has a computation task to be executed. Otherwise, the mobile user is inactive. We use the on-off distribution to model mobile user i's activeness, i.e., mobile user i is active (or inactive) with probability θ_i (or $1-\theta_i$).

To model the time-varying wireless channels, we assume that the channels between mobile users and the AP follow Rayleigh fading, which is a realistic and widely adopted mobile channel model [33], [34]. Specifically, the instantaneous channel power gain from user i to the AP is given by $g_{i,o} = (d_{i,o})^{-\alpha}\,\beta_{i,o}$, where $d_{i,o}$ is the distance from user i to the AP (for clear presentation, we use "o" to denote the AP), α is the path loss exponent, and $\beta_{i,o}$ is the Rayleigh fading factor. Notice that, the instantaneous random coefficient $\beta_{i,o}$ varies from time slot to time slot.

We consider a more practical model that all system parameters (i.e., channel power gains and the users' active probabilities) are unknown. For convenience of analysis, we define a probability space as $(\Omega, \mathcal{H}, \mathbb{P})$, where Ω is the sample space over all system states, $\mathcal{H}$ is a minimal σ-algebra on subsets of Ω, and $\mathbb{P}$ is a probability measure on $(\Omega, \mathcal{H})$. Let Λ denote an event in the sample space Ω. $\Theta(\Lambda) = [\boldsymbol{a}(\Lambda), \mathbf{g}(\Lambda)] : \Omega \to 2^N \times \mathbb{R}^N$ is a random vector, where $\boldsymbol{a} = [a_i]_{\forall i \in \mathcal{N}}$, $a_i \in \{0, 1\}$ denotes the user i's state (0 for inactive, and 1 for active) that satisfies the on-off distribution with probability θ_i, and $\mathbf{g} = [g_{i,o}]_{\forall i \in \mathcal{N}}$ follows Rayleigh fading. For better reading, Table 1 summarizes the mainly used notations in this paper.

3.2 Communication Model for Active Mobile Users

To make a clear presentation, we first consider one realization of the stochastic system state, which is denoted by Λ. Given Λ, we define the set of active users as $\mathcal{A} = \{i \in \mathcal{N} : a_i = 1\}$. Suppose that user i chooses to offload its computation task to the cloud via wireless channel $s_i > 0$. Given the strategy profile $\boldsymbol{s}_{\mathcal{A}} = [s_i]_{\forall i \in \mathcal{A}}$ of all active mobile users, the uplink data rate of user $i \in \mathcal{A}$ can be computed by[2]

$$R_i(\boldsymbol{s}_{\mathcal{A}}, \Lambda) = B\log_2\left(1 + \frac{p_i g_{i,o}}{\sum\limits_{j \in \mathcal{A}\backslash\{i\}: s_j = s_i} p_j g_{j,o} + \sigma_0}\right), \quad (1)$$

where B is the channel bandwidth, p_i is the transmit power of user i, $g_{i,o}$ is the channel gain from user i to the AP, and σ_0 denotes the background noise power. As shown in (1), if too many active users choose the same channel to offload their computation tasks to the AP, they may incur severe interference to each other, resulting in low offloading rates.

3.3 Computation Model for Active Mobile Users

We consider that each active user $i \in \mathcal{A}$ has a computation task $\mathcal{T}_i$ that needs to be executed either locally on the mobile device or remotely on the telecom cloud through computation offloading. The computation task can be expressed as $\mathcal{T}_i = \left(C_i, D_i^{\text{loc}}, D_i^{\text{clo}}\right)$, where C_i is the size of all input computation data (e.g., the mobile system settings, the program codes, and the input parameters) involved in the task, D_i^{loc} and D_i^{clo} are the total number of CPU cycles required to complete the computation task on the mobile device and the telecom cloud, respectively[3]. For the sake of clear presentation, we use the letters "loc" and "clo" to represent LOCal computing and CLOud computing, respectively. The detailed modelings of the computation cost are as follows.

3.3.1 Computation Cost When Choosing to Perform Cloud Computing

If active user i chooses to offload its computation task $\mathcal{T}_i$ to the cloud, it would incur the cost for transmitting the input data to the

2. In this paper, we focus on the wireless interference model given in (1) which is widely adopted in the literature. Note that, we can also adopt some media access control protocols (such as CSMA) in which the multiple access among users for the shared spectrum is carried out over the packet level. As shown in [11], the analysis could be very similar to our adopted wireless interference model.

3. A mobile user i can obtain the information of C_i, D_i^{loc}, and D_i^{clo} by applying the methods (e.g., call graph analysis) in [35], [36]. If we consider the related overhead, it would be added just as a constant in the system model, which would not affect the following mathematical analysis. Besides, since the mobile devices and the remote cloud computing servers have different instruction set architectures, the numbers of CPU cycles for the two architectures (i.e., D_i^{loc} and D_i^{clo}) are different.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2018.2847337, IEEE Transactions on Mobile Computing

cloud via wireless access. According to the communication model (1), the transmission time and energy consumption for offloading the input data of size C_i can be respectively computed by

$$T_{i,1}^{\text{clo}}\left(\boldsymbol{s}_{\mathcal{A}},\Lambda\right)=\frac{C_i}{R_i\left(\boldsymbol{s}_{\mathcal{A}},\Lambda\right)},\ \text{ and }\ E_i^{\text{clo}}\left(\boldsymbol{s}_{\mathcal{A}},\Lambda\right)=\frac{p_iC_i}{R_i\left(\boldsymbol{s}_{\mathcal{A}},\Lambda\right)}.$$

After the transmission, the cloud expends the execution time $T_{i,2}^{\text{clo}}=\frac{D_i^{\text{clo}}}{F_i^{\text{clo}}}$ to finish mobile user i's task, where F_i^{clo} denotes the computation capability (i.e., CPU cycles per second) assigned to user i by the cloud[4]. Then, considering both the *processing time*[5] and the *energy consumption*, the total cost[6] when mobile user i chooses to perform cloud computing (i.e., $s_i>0$) can be given by:

$$\begin{aligned}V_i^{\text{clo}}\left(\boldsymbol{s}_{\mathcal{A}},\Lambda\right)&=\mu_i^T\left(T_{i,1}^{\text{clo}}\left(\boldsymbol{s}_{\mathcal{A}},\Lambda\right)+T_{i,2}^{\text{clo}}\right)+\mu_i^E E_i^{\text{clo}}\left(\boldsymbol{s}_{\mathcal{A}},\Lambda\right)\\&=\frac{\left(\mu_i^E p_i+\mu_i^T\right)C_i}{R_i\left(\boldsymbol{s}_{\mathcal{A}},\Lambda\right)}+\mu_i^T T_{i,2}^{\text{clo}},\qquad i\in\mathcal{A},\text{and } s_i>0,\end{aligned}\tag{2}$$

where μ_i^T and $\mu_i^E\in(0,1)$ denote the weights of computational time and energy for mobile user i's strategy decision, respectively. Here, the units of μ_i^T and μ_i^E are $\frac{1}{\text{Second}}$ and $\frac{1}{\text{Joule}}$, respectively.

3.3.2 Computation Cost When Choosing to Perform Local Computing

Each active mobile user i can also choose to execute its computation task $\mathcal{T}_i$ locally by itself (i.e., without invoking any computation offloading to the cloud). Let F_i^{loc} be the computation capability of mobile user i. The computation execution time of task $\mathcal{T}_i$ by local computing is then given by $T_i^{\text{loc}}=\frac{D_i^{\text{loc}}}{F_i^{\text{loc}}}$. The corresponding computational energy can be computed by $E_i^{\text{loc}}=\eta_i D_i^{\text{loc}}$, where η_i is the coefficient denoting the energy consumption per CPU cycle. Then, by taking into account both the *processing time* and the *energy consumption*, we can compute mobile user i's total cost when it chooses to perform local computation (i.e., $s_i=0$) as follows:

$$V_i^{\text{loc}}=\mu_i^T T_i^{\text{loc}}+\mu_i^E E_i^{\text{loc}},\qquad i\in\mathcal{A},\text{ and } s_i=0.\tag{3}$$

Based on the communication and computation models above, we see that, when choosing to offload computation task to the cloud, each mobile user's cost depends not only on its own offloading strategy, but also on all the other active peers'. Specifically, as shown in the cost function (2), if too many mobile users are active and using the same strategy (i.e., choosing to use the same wireless channel) to offload their computation tasks to the cloud, they may experience low offloading rates, which will incur more computation cost (including longer transmission time and higher energy consumption). In this case, it would be more beneficial for the mobile user to compute the task locally by itself. Due to such an inter-dependence among different mobile users, game theory is a suitable mathematical tool to model and analyze users' decision making for computation offloading. However, due to dynamically varying of mobile users' activeness, each user may not be able to know other peers' active/inactive states. Moreover, mobile users prefer better channel condition for offloading their computation tasks, but the wireless channels are time-varying, which makes the problem more challenging.

4. F_i^{clo} is considered to be preset for each user. In this paper, we do not study the allocation/scheduling of computation resources to different users from the perspective of mobile cloud, since this issue has been widely investigated in literature, e.g., [4], [20].

5. Since the AP is connected to the mobile cloud via the high-speed fiber link, the transmission time cost among them could be neglected, compared with the much higher wireless access time cost (resulted from the constrained wireless spectrum resource). Besides, since the size of the computation outcome is usually much smaller than the size of input computation data for many applications (e.g., face recognition), we neglect the time cost for the cloud to send the computation outcome back to the mobile user. The similar assumption also appears in many previous works [11], [17], [18], [19], [21].

6. It would also be interesting to consider users' economic cost. Since F_i^{clo} is considered to be preset for each user in our system model, the economic cost is just a constant added into Eq. (2), and thus it will not affect the following game-theoretic solution. However, if F_i^{clo} is considered as an optimization variable, how to optimally decide the price for the computing resources would also be a key problem for the service providers. In this case, the economic game models such as the market model, bargaining model, bidding model, auction model, duopoly model, and Stackelberg model could be adopted. Since this economic consideration will lead to significant change of the current game model, we consider this issue as an important future direction to extend our work.

4 Stochastic Computation Offloading Game

In this section, we formulate a stochastic game to model the decision process for mobile users' computation offloading to the cloud. For the sake of clear presentation, we first describe the game model in a static case, and then illustrate the dynamic case.

4.1 Game Models

4.1.1 Static Case

Given other users' strategy decisions, each active user $i\in\mathcal{A}$ independently adjusts its computation offloading strategy to minimize its own computation cost. Specifically, given a realization Λ in the probability space $(\Omega,\mathcal{H},\mathbb{P})$, the (state-based) payoff function of each active user i is naturally defined by

$$u_i^0\left(s_i,\boldsymbol{s}_{\mathcal{A}\backslash\{i\}},\Lambda\right)=\begin{cases}V_i^{\text{loc}}, & if\ \ s_i=0;\\ V_i^{\text{clo}}\left(s_i,\boldsymbol{s}_{\mathcal{A}\backslash\{i\}},\Lambda\right), & if\ \ s_i>0,\end{cases}\tag{4}$$

where s_i denotes active user i's strategy, $\boldsymbol{s}_{\mathcal{A}\backslash\{i\}}$ denotes the strategy profile of all the active users excluding user i, V_i^{clo} and V_i^{loc} are defined in (2) and (3), respectively. Then, the game can be formulated as $\mathcal{G}_{0,\Lambda}=\left[\mathcal{A},\Lambda,\{\mathcal{S}_i\}_{i\in\mathcal{N}},\{u_i^0\}_{i\in\mathcal{N}}\right]$, where $\mathcal{A}$ is the set of active users, $\mathcal{S}_i$ denotes active user i's strategy space, $\mathcal{S}_i=\{0,1,\ldots,M\}$. Each active user autonomously chooses its strategy s_i to minimize its own payoff, i.e.,

$$(\mathcal{G}_{0,\Lambda}):\quad \min_{s_i\in\mathcal{S}_i} u_i^0\left(s_i,\boldsymbol{s}_{\mathcal{A}\backslash\{i\}},\Lambda\right),\forall i\in\mathcal{A}.\tag{5}$$

Definition 1: Given a computation offloading strategy profile $\boldsymbol{s}_{\mathcal{A}}=[s_i]_{\forall i\in\mathcal{A}}$, for an active user $i\in\mathcal{A}$ that chooses the cloud computing approach (i.e., $s_i>0$), if the cloud computing does not yield a higher cost than the local computing (i.e., $V_i^{\text{clo}}\left(\boldsymbol{s}_{\mathcal{A}},\Lambda\right)\leq V_i^{\text{loc}}$), we say that the cloud computing is beneficial to user i.

In particular, from (1), (2), and (3), we observe that whether offloading computation task to cloud is beneficial to mobile user i or not strongly depends on its suffered interference, i.e., $I_i\left(\boldsymbol{s}_{\mathcal{A}},\Lambda\right)=\sum_{j\in\mathcal{A}\backslash\{i\}:s_j=s_i}p_j g_{j,o}$. Referring to the similar proof in [11], we can achieve Lemma 1 as follows:

Lemma 1. *Given a computation offloading strategy profile $\boldsymbol{s}_{\mathcal{A}}$, cloud computing is beneficial to an active user $i\in\mathcal{A}$ if its suffered interference $I_i(\boldsymbol{s}_{\mathcal{A}},\Lambda)=\sum_{j\in\mathcal{A}\backslash\{i\}:s_j=s_i}p_j g_{j,o}$ on the selected wireless channel $s_i>0$ satisfies that $I_i\left(\boldsymbol{s}_{\mathcal{A}},\Lambda\right)\leq Q_i$, with the threshold $Q_i=\frac{p_i g_{i,o}}{2^{\psi_i}-1}-\sigma_0$, where the parameter $\psi_i=\frac{\left(\mu_i^E p_i+\mu_i^T\right)C_i}{B\left(\mu_i^T T_i^{\text{loc}}+\mu_i^E E_i^{\text{loc}}-\mu_i^T T_{i,2}^{\text{clo}}\right)}$.*

Proof: According to Definition 1, cloud computing is beneficial to user i only if $V_i^{\text{clo}}\left(\boldsymbol{s}_{\mathcal{A}},\Lambda\right)\leq V_i^{\text{loc}}$. Based on the computing models in (2) and (3), this condition corresponds to

$$\frac{\left(\mu_i^E p_i+\mu_i^T\right)C_i}{R_i\left(\boldsymbol{s}_{\mathcal{A}},\Lambda\right)}+\mu_i^T T_{i,2}^{\text{clo}}\leq\mu_i^T T_i^{\text{loc}}+\mu_i^E E_i^{\text{loc}},$$

which after some manipulations leads to the following condition:

$$R_i\left(\boldsymbol{s}_{\mathcal{A}},\Lambda\right)\geq\frac{\left(\mu_i^E p_i+\mu_i^T\right)C_i}{\mu_i^T T_i^{\text{loc}}+\mu_i^E E_i^{\text{loc}}-\mu_i^T T_{i,2}^{\text{clo}}}.$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2018.2847337, IEEE Transactions on Mobile Computing

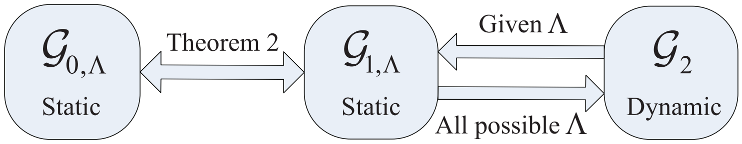


Fig. 2. Graphical representation of relationships among the three proposed game models.

Based on the communication model (1), the above condition can be equivalently transformed into:

$$\sum\nolimits_{j\in\mathcal{A}\setminus\{i\}:s_j=s_i} p_j g_{j,o} \leq \frac{p_i g_{i,o}}{2^{\frac{(\mu_i^E p_i+\mu_i^T)C_i}{B\left(\mu_i^T T_i^{\text{loc}}+\mu_i^E E_i^{\text{loc}}-\mu_i^T T_{i,2}^{\text{clo}}\right)}}-1}-\sigma_0.$$

This completes the proof of Lemma 1. □

Based on Lemma 1, we design the following game model $\mathcal{G}_{1,\Lambda}$ (which will be shown to be equivalent to $\mathcal{G}_{0,\Lambda}$ with proofs in Lemma 2 and Theorem 2) as follows:

$$(\mathcal{G}_{1,\Lambda}): \quad \min_{s_i\in\mathcal{S}_i} u_i\left(s_i, \boldsymbol{s}_{\mathcal{A}\setminus\{i\}}, \Lambda\right), \forall i\in\mathcal{A}, \tag{6}$$

where each user's payoff function is given by:

$$u_i\left(s_i, \boldsymbol{s}_{\mathcal{A}\setminus\{i\}}, \Lambda\right) = \begin{cases} Q_i, & if \ \ s_i=0; \\ I_i\left(\boldsymbol{s}_{\mathcal{A}}, \Lambda\right), & if \ \ s_i>0. \end{cases} \tag{7}$$

4.1.2 Dynamic Case

We next extend the static game $\mathcal{G}_{1,\Lambda}$ (under a given Λ) into a corresponding stochastic game that experiences all possible Λ. Specifically, in the dynamic and stochastic environment, we define an *expected payoff function* for each user $i\in\mathcal{N}$ as follows:

$$\bar{u}_i(s_i, \boldsymbol{s}_{-i}) = \mathbb{E}_\Theta[u_i\left(s_i, \boldsymbol{s}_{-i}, \Theta\right)] = \begin{cases} \bar{Q}_i, & if \ s_i=0; \\ \mathbb{E}_\Theta[I_i\left(s_i, \boldsymbol{s}_{-i}, \Theta\right)], & if \ s_i>0, \end{cases} \tag{8}$$

where $\boldsymbol{s}_{-i}=[s_j]_{\forall j\in\mathcal{N}\setminus\{i\}}$ denotes the strategy profile of all users excluding user i, $\bar{Q}_i=\frac{p_i\bar{g}_{i,o}}{2^{\psi_i}-1}-\sigma_0$, and $\bar{g}_{i,o}$ is the expected channel gain from mobile user i to the AP. Based on (8), we formulate a stochastic game denoted by $\mathcal{G}_2=\left[\mathcal{N}, \Theta, \{\mathcal{S}_i\}_{i\in\mathcal{N}}, \{\bar{u}_i\}_{i\in\mathcal{N}}\right]$. Each mobile user independently adjusts its strategy to minimize its individual expected payoff function, which can be expressed as:

$$(\mathcal{G}_2): \quad \min_{s_i\in\mathcal{S}_i} \bar{u}_i\left(s_i, \boldsymbol{s}_{-i}\right), \forall i\in\mathcal{N}. \tag{9}$$

Figure 2 illustrates the connections among the game models $\mathcal{G}_{0,\Lambda}$, $\mathcal{G}_{1,\Lambda}$, and $\mathcal{G}_2$.

4.2 Analysis of Nash Equilibrium

In game theory, Nash Equilibrium (NE) is the most important solution concept for analyzing the outcome of the strategic interaction of multiple decision-makers. In this subsection, we first investigate the existence of NE for the static game models $\mathcal{G}_{0,\Lambda}$ and $\mathcal{G}_{1,\Lambda}$. Moreover, by proving the equivalence between $\mathcal{G}_{0,\Lambda}$ and $\mathcal{G}_{1,\Lambda}$, we illustrate the rationality of designing the game model $\mathcal{G}_{1,\Lambda}$. Then, based on the analysis in the static case, we derive the existence of NE in the stochastic game $\mathcal{G}_2$. To proceed, we first introduce the definition of NE for game models in both static and dynamic cases.

Definition 2 (NE of $\mathcal{G}_{0,\Lambda}$): For a realization $\Lambda\in\Omega$, a computation offloading strategy profile $\boldsymbol{s}_{\mathcal{A}}^*=[s_i^*]_{\forall i\in\mathcal{A}}$ is a (pure-strategy) NE of the game $\mathcal{G}_{0,\Lambda}$ if and only if no active mobile user can minimize its payoff function u_i^0 by unilaterally deviating, i.e.,

$$u_i^0\left(s_i^*, \boldsymbol{s}_{\mathcal{A}\setminus\{i\}}^*, \Lambda\right) \leq u_i^0\left(s_i, \boldsymbol{s}_{\mathcal{A}\setminus\{i\}}^*, \Lambda\right), \forall i\in\mathcal{A}, \forall s_i\in\mathcal{S}_i. \tag{10}$$

Definition 3 (NE of $\mathcal{G}_{1,\Lambda}$): For a realization $\Lambda\in\Omega$, a computation offloading strategy profile $\boldsymbol{s}_{\mathcal{A}}^*=[s_i^*]_{\forall i\in\mathcal{A}}$ is a (pure-strategy) NE of the game $\mathcal{G}_{1,\Lambda}$ if and only if no active mobile user can minimize its payoff function u_i by unilaterally deviating, i.e.,

$$u_i\left(s_i^*, \boldsymbol{s}_{\mathcal{A}\setminus\{i\}}^*, \Lambda\right) \leq u_i\left(s_i, \boldsymbol{s}_{\mathcal{A}\setminus\{i\}}^*, \Lambda\right), \forall i\in\mathcal{A}, \forall s_i\in\mathcal{S}_i. \tag{11}$$

Definition 4 (Expected NE of $\mathcal{G}_2$): A computation offloading strategy profile $\boldsymbol{s}^*=[s_i^*]_{\forall i\in\mathcal{N}}$ is an expected (pure-strategy) NE of the stochastic game $\mathcal{G}_2$ if and only if no mobile user can minimize its expected payoff function $\bar{u}_i$ by deviating unilaterally, i.e.,

$$\bar{u}_i\left(s_i^*, \boldsymbol{s}_{-i}^*\right) \leq \bar{u}_i\left(s_i, \boldsymbol{s}_{-i}^*\right), \forall i\in\mathcal{N}, \forall s_i\in\mathcal{S}_i. \tag{12}$$

Theorem 1. *For an arbitrary realization $\Lambda\in\Omega$, $\mathcal{G}_{1,\Lambda}$ is a weighted potential game which has at least one NE.*

Proof: The key of the proof is to show that for each user $k\in\mathcal{A}$, the change of its payoff function (due to its unilateral change of strategy) is proportional to the change in a carefully chosen potential function for the whole system. The details are as follows.

We first construct a state-based potential function as follows:

$$\begin{aligned}\Phi\left(\boldsymbol{s}_{\mathcal{A}}, \Lambda\right) = &\frac{1}{2}\sum_{i\in\mathcal{A}}\sum_{j\in\mathcal{A}\setminus\{i\}} p_i g_{i,o} p_j g_{j,o} \ell_{\{s_j=s_i\}} \ell_{\{s_i>0\}} \\ &+ \sum_{i\in\mathcal{A}} p_i g_{i,o} Q_i \ell_{\{s_i=0\}},\end{aligned} \tag{13}$$

where $\ell_{\{condition\}}$ is an indicator function, and it is equal to 0 (resp., 1) when the *condition* is false (resp., true). The above equation (13) can be equivalently written as follows:

$$\begin{aligned}\Phi\left(\boldsymbol{s}_{\mathcal{A}}, \Lambda\right) = &\frac{1}{2}\Bigg(\sum_{j\in\mathcal{A}\setminus\{k\}} p_k g_{k,o} p_j g_{j,o} \ell_{\{s_j=s_k\}} \ell_{\{s_k>0\}} \\ &+ \sum_{i\in\mathcal{A}\setminus\{k\}} p_i g_{i,o} p_k g_{k,o} \ell_{\{s_k=s_i\}} \ell_{\{s_i>0\}} \\ &+ \sum_{i\in\mathcal{A}\setminus\{k\}}\sum_{j\in\mathcal{A}\setminus\{i,k\}} p_i g_{i,o} p_j g_{j,o} \ell_{\{s_j=s_i\}} \ell_{\{s_i>0\}}\Bigg) \\ &+ p_k g_{k,o} Q_k \ell_{\{s_k=0\}} + \sum_{i\in\mathcal{A}\setminus\{k\}} p_i g_{i,o} Q_i \ell_{\{s_i=0\}}.\end{aligned} \tag{14}$$

In particular, the following result always holds:

$$\sum_{j\in\mathcal{A}\setminus\{k\}} p_k g_{k,o} p_j g_{j,o} \ell_{\{s_j=s_k\}} \ell_{\{s_k>0\}} = \sum_{i\in\mathcal{A}\setminus\{k\}} p_i g_{i,o} p_k g_{k,o} \ell_{\{s_k=s_i\}} \ell_{\{s_i>0\}}. \tag{15}$$

Using (14) and (15), we can derive the following result:

$$\begin{aligned}\Phi\left(\boldsymbol{s}_{\mathcal{A}}, \Lambda\right) = &\sum_{j\in\mathcal{A}\setminus\{k\}} p_k g_{k,o} p_j g_{j,o} \ell_{\{s_j=s_k\}} \ell_{\{s_k>0\}} \\ &+ p_k g_{k,o} Q_k \ell_{\{s_k=0\}} + \Xi\left(\boldsymbol{s}_{\mathcal{A}\setminus\{k\}}, \Lambda\right),\end{aligned} \tag{16}$$

where $\Xi\left(\boldsymbol{s}_{\mathcal{A}\setminus\{k\}}, \Lambda\right) = \frac{1}{2}\sum_{i\in\mathcal{A}\setminus\{k\}}\sum_{j\in\mathcal{A}\setminus\{i,k\}} p_i g_{i,o} p_j g_{j,o} \ell_{\{s_j=s_i\}} \ell_{\{s_i>0\}} + \sum_{i\in\mathcal{A}\setminus\{k\}} p_i g_{i,o} Q_i \ell_{\{s_i=0\}}$ is independent of user k's strategy s_k.

Besides, based on (7), we have the following equation:

$$\begin{aligned}u_k\left(s_k, \boldsymbol{s}_{\mathcal{A}\setminus\{k\}}, \Lambda\right) &= I_k\left(\boldsymbol{s}_{\mathcal{A}}, \Lambda\right)\ell_{\{s_k>0\}} + Q_k \ell_{\{s_k=0\}} \\ &= \sum_{j\in\mathcal{A}\setminus\{k\}} p_j g_{j,o} \ell_{\{s_j=s_k\}} \ell_{\{s_k>0\}} + Q_k \ell_{\{s_k=0\}}.\end{aligned} \tag{17}$$

Therefore, for each user $k \in \mathcal{A}$ and its two different strategies s_k and s'_k, we have the following equation:

$$\begin{aligned}&\Phi\left(s'_k, \boldsymbol{s}_{\mathcal{A}\setminus\{k\}}, \Lambda\right) - \Phi\left(s_k, \boldsymbol{s}_{\mathcal{A}\setminus\{k\}}, \Lambda\right)\\ &= \sum_{j\in\mathcal{A}\setminus\{k\}} p_k g_{k,o} p_j g_{j,o} \ell_{\{s_j = s'_k\}} \ell_{\{s'_k>0\}} + p_k g_{k,o} Q_k \ell_{\{s'_k=0\}}\\ &\quad - \sum_{j\in\mathcal{A}\setminus\{k\}} p_k g_{k,o} p_j g_{j,o} \ell_{\{s_j = s_k\}} \ell_{\{s_k>0\}} - p_k g_{k,o} Q_k \ell_{\{s_k=0\}}\\ &= p_k g_{k,o}\Big(u_k\left(s'_k, \boldsymbol{s}_{\mathcal{A}\setminus\{k\}}, \Lambda\right) - u_k\left(s_k, \boldsymbol{s}_{\mathcal{A}\setminus\{k\}}, \Lambda\right)\Big).\end{aligned} \tag{18}$$

As stated before, (18) essentially means that for each user $k \in \mathcal{A}$, the change of its payoff function (due to its unilateral change of strategy) is proportional to the change in the potential function (13) for the whole system. Thus, according to the potential game theory in [24], $\mathcal{G}_{1,\Lambda}$ is a weighted potential game (with weight-factor $p_k g_{k,o}$) which has at least one NE. This concludes the proof. □

According to the proof in [11], $\mathcal{G}_{0,\Lambda}$ is an ordinal potential game which also has at least one pure-strategy NE point. In the following, we will investigate the relationship between $\mathcal{G}_{0,\Lambda}$ and $\mathcal{G}_{1,\Lambda}$, and illustrate the rationality of designing the game model $\mathcal{G}_{1,\Lambda}$.

Lemma 2. *In $\mathcal{G}_{0,\Lambda}$ and $\mathcal{G}_{1,\Lambda}$, all users' strategy preferences are the same. That is, for an arbitrary realization $\Lambda \in \Omega$, $\forall i \in \mathcal{A}$, for any $s'_i \neq s_i$,*

$$u_i^0(s'_i, s_{\mathcal{A}\setminus\{i\}}, \Lambda) \le u_i^0(s_i, s_{\mathcal{A}\setminus\{i\}}, \Lambda) \Leftrightarrow u_i(s'_i, s_{\mathcal{A}\setminus\{i\}}, \Lambda) \le u_i(s_i, s_{\mathcal{A}\setminus\{i\}}, \Lambda). \tag{19}$$

Proof: The proof is essentially based on our previous Lemma 1. Specifically, we consider the following three cases:

1) $s'_i > 0$, $s_i = 0$: According to the definition of payoff functions, $u_i^0(s'_i, s_{\mathcal{A}\setminus\{i\}}, \Lambda) = V_i^{\text{clo}}(s'_i, s_{\mathcal{A}\setminus\{i\}}, \Lambda)$, $u_i^0(s_i, s_{\mathcal{A}\setminus\{i\}}, \Lambda) = V_i^{\text{loc}}$, and $u_i(s'_i, s_{\mathcal{A}\setminus\{i\}}, \Lambda) = I_i(s'_i, s_{\mathcal{A}\setminus\{i\}}, \Lambda)$, $u_i(s_i, s_{\mathcal{A}\setminus\{i\}}, \Lambda) = Q_i$. Based on Lemma 1, $V_i^{\text{clo}}(s'_i, s_{\mathcal{A}\setminus\{i\}}, \Lambda) \le V_i^{\text{loc}} \Leftrightarrow I_i(s'_i, s_{\mathcal{A}\setminus\{i\}}, \Lambda) \le Q_i$. Therefore, $u_i^0(s'_i, s_{\mathcal{A}\setminus\{i\}}, \Lambda) \le u_i^0(s_i, s_{\mathcal{A}\setminus\{i\}}, \Lambda) \Leftrightarrow u_i(s'_i, s_{\mathcal{A}\setminus\{i\}}, \Lambda) \le u_i(s_i, s_{\mathcal{A}\setminus\{i\}}, \Lambda)$.
2) $s'_i = 0$, $s_i > 0$: According to the definition of payoff functions, $u_i^0(s'_i, s_{\mathcal{A}\setminus\{i\}}, \Lambda) = V_i^{\text{loc}}$, $u_i^0(s_i, s_{\mathcal{A}\setminus\{i\}}, \Lambda) = V_i^{\text{clo}}(\boldsymbol{s}_{\mathcal{A}}, \Lambda)$, and $u_i(s'_i, s_{\mathcal{A}\setminus\{i\}}, \Lambda) = Q_i$, $u_i(s_i, s_{\mathcal{A}\setminus\{i\}}, \Lambda) = I_i(\boldsymbol{s}_{\mathcal{A}}, \Lambda)$. Based on Lemma 1, $V_i^{\text{clo}}(\boldsymbol{s}_{\mathcal{A}}, \Lambda) \ge V_i^{\text{loc}} \Leftrightarrow I_i(\boldsymbol{s}_{\mathcal{A}}, \Lambda) \ge Q_i$. Therefore, $u_i^0(s'_i, s_{\mathcal{A}\setminus\{i\}}, \Lambda) \le u_i^0(s_i, s_{\mathcal{A}\setminus\{i\}}, \Lambda) \Leftrightarrow u_i(s'_i, s_{\mathcal{A}\setminus\{i\}}, \Lambda) \le u_i(s_i, s_{\mathcal{A}\setminus\{i\}}, \Lambda)$.
3) $s'_i > 0$, $s_i > 0$: According to the definition of payoff functions, $u_i^0(s'_i, s_{\mathcal{A}\setminus\{i\}}, \Lambda) = V_i^{\text{clo}}(s'_i, s_{\mathcal{A}\setminus\{i\}}, \Lambda)$, $u_i^0(s_i, s_{\mathcal{A}\setminus\{i\}}, \Lambda) = V_i^{\text{clo}}(\boldsymbol{s}_{\mathcal{A}}, \Lambda)$, and $u_i(s'_i, s_{\mathcal{A}\setminus\{i\}}, \Lambda) = I_i(s'_i, s_{\mathcal{A}\setminus\{i\}}, \Lambda)$, $u_i(s_i, s_{\mathcal{A}\setminus\{i\}}, \Lambda) = I_i(\boldsymbol{s}_{\mathcal{A}}, \Lambda)$. Since $V_i^{\text{clo}} = \frac{(\mu_i^E p_i + \mu_i^T) C_i}{R_i} + \mu_i^T T_{i,2}^{\text{clo}}$, we can get the following result: $V_i^{\text{clo}}(s'_i, s_{\mathcal{A}\setminus\{i\}}, \Lambda) \le V_i^{\text{clo}}(\boldsymbol{s}_{\mathcal{A}}, \Lambda) \Leftrightarrow R_i(s'_i, s_{\mathcal{A}\setminus\{i\}}, \Lambda) \ge R_i(\boldsymbol{s}_{\mathcal{A}}, \Lambda)$. As R_i is a decreasing function of the received interference I_i, $R_i(s'_i, s_{\mathcal{A}\setminus\{i\}}, \Lambda) \ge R_i(\boldsymbol{s}_{\mathcal{A}}, \Lambda) \Leftrightarrow I_i(s'_i, s_{\mathcal{A}\setminus\{i\}}, \Lambda) \le I_i(\boldsymbol{s}_{\mathcal{A}}, \Lambda)$. Therefore, $u_i^0(s'_i, s_{\mathcal{A}\setminus\{i\}}, \Lambda) \le u_i^0(s_i, s_{\mathcal{A}\setminus\{i\}}, \Lambda) \Leftrightarrow u_i(s'_i, s_{\mathcal{A}\setminus\{i\}}, \Lambda) \le u_i(s_i, s_{\mathcal{A}\setminus\{i\}}, \Lambda)$.

By summarizing the above three cases, we can obtain the results in Lemma 2. □

Theorem 2. *Each NE of game $\mathcal{G}_{0,\Lambda}$ is an NE of game $\mathcal{G}_{1,\Lambda}$, and each NE of game $\mathcal{G}_{1,\Lambda}$ is also an NE of game $\mathcal{G}_{0,\Lambda}$.*

Proof: Denote the set of NE of $\mathcal{G}_{0,\Lambda}$ by Ψ_0, and the set of NE of $\mathcal{G}_{1,\Lambda}$ by Ψ_1. $\forall \mathbf{s}^* \in \Psi_0$, according to the definition of NE, $u_i^0\left(s_i^*, s^*_{\mathcal{A}\setminus\{i\}}, \Lambda\right) \le u_i^0\left(s_i, s^*_{\mathcal{A}\setminus\{i\}}, \Lambda\right), \forall i \in \mathcal{A}, \forall s_i \in \mathcal{S}_i$. Then, based on (19), we have $u_i\left(s_i^*, s^*_{\mathcal{A}\setminus\{i\}}, \Lambda\right) \le u_i\left(s_i, s^*_{\mathcal{A}\setminus\{i\}}, \Lambda\right), \forall i \in \mathcal{A}, \forall s_i \in \mathcal{S}_i$, which implies that $\mathbf{s}^* \in \Psi_1$. Following the similar argument, we also have $\forall \mathbf{s}^* \in \Psi_1 \Rightarrow \mathbf{s}^* \in \Psi_0$. Therefore, we can conclude that Ψ_0 and Ψ_1 are the same. □

Lemma 2 and Theorem 2 together mean that $\mathcal{G}_{0,\Lambda}$ and $\mathcal{G}_{1,\Lambda}$ are essentially equivalent. Thus, by deriving the NE of $\mathcal{G}_{1,\Lambda}$, we can also get the NE for $\mathcal{G}_{0,\Lambda}$. Moreover, $\mathcal{G}_{1,\Lambda}$ can be used for designing the stochastic game model $\mathcal{G}_2$. Based on the property of $\mathcal{G}_{1,\Lambda}$ as shown in Theorem 1, we can obtain the following result.

Theorem 3. *The stochastic game $\mathcal{G}_2$ is a weighted potential game with the expected potential function given by:*

$$\begin{aligned}&\bar{\Phi}(\boldsymbol{s}) = \mathbb{E}_{\Theta}[\Phi(\boldsymbol{s}, \Theta)]\\ &= \frac{1}{2}\sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}\setminus\{i\}} \theta_i\theta_j p_i \bar{g}_{i,o} p_j \bar{g}_{j,o} \ell_{\{s_j=s_i\}} \ell_{\{s_i>0\}} + \sum_{i\in\mathcal{N}} \theta_i p_i \bar{g}_{i,o} \bar{Q}_i \ell_{\{s_i=0\}},\end{aligned} \tag{20}$$

where $\boldsymbol{s} = [s_i]_{\forall i\in\mathcal{N}}$ denotes the strategy profile of all mobile users, θ_i is the active probability of mobile user $i \in \mathcal{N}$.

Proof: By taking the operation of expectation for (18), we can obtain the following result

$$\bar{\Phi}(s'_k, \boldsymbol{s}_{-k}) - \bar{\Phi}(s_k, \boldsymbol{s}_{-k}) = p_k \bar{g}_{k,o}\Big(\bar{u}_k(s'_k, \boldsymbol{s}_{-k}) - \bar{u}_k(s_k, \boldsymbol{s}_{-k})\Big), \tag{21}$$

where $\bar{u}_k(s_k, \boldsymbol{s}_{-k})$ is the payoff function defined in (8). Thus, according to the potential game theory in [24], $\mathcal{G}_2$ is a weighted potential game with weight-factor $p_k \bar{g}_{k,o}$. □

As proved in [24], every weighted potential game possesses the finite improvement property, and thus $\mathcal{G}_2$ has at least one pure-strategy NE. That is, the investigated mobile users' decision-making process for computation offloading is guaranteed to have a pure-strategy NE under the dynamic environment. However, the behaviors of users in the game are selfish to minimize its own payoff (without caring about the other peers'), which may lead to an inefficient NE. In the following section, we will analyze the achievable performance of NE for the stochastic game $\mathcal{G}_2$ under dynamic environment.

5 Performance Analysis of Nash Equilibrium

To evaluate the performance of the NE, we first study the metric of system-wide computation cost, and then analyze the number of mobile users who benefit from cloud computing.

5.1 Metric I: System-Wide Computation Cost

In dynamic and stochastic environment, (2) cannot characterize the cost of cloud computing, since the transmission rate $R_i(\boldsymbol{s})$ is dynamically varying. Therefore, we compute the expected cost of offloading computation task to mobile cloud in dynamic environment as follows[7]:

$$\bar{V}_i^{\text{clo}}(\boldsymbol{s}) = \frac{\left(\mu_i^E p_i + \mu_i^T\right) C_i}{\mathbb{E}_{\Theta}[R_i(\boldsymbol{s}, \Theta)]} + \mu_i^T T_{i,2}^{\text{clo}}. \tag{22}$$

In comparison, each mobile user's total cost for performing local computing is still given by (3), since the dynamic environment (channel dynamics, users' dynamic activeness) does not impact the local computing. In summary, we evaluate the computation cost in dynamic environment by the following metric:

$$\Gamma_i(s_i, \boldsymbol{s}_{-i}) = \begin{cases} V_i^{\text{loc}}, & if \quad s_i = 0;\\ \bar{V}_i^{\text{clo}}(\boldsymbol{s}), & if \quad s_i > 0.\end{cases} \tag{23}$$

7. An alternative choice of the expected cost can be $\bar{V}_i^{\text{clo}}(\boldsymbol{s}) = \mathbb{E}_{\Theta}[\frac{(\mu_i^E p_i + \mu_i^T) C_i}{R_i(\boldsymbol{s},\Theta)}] + \mu_i^T T_{i,2}^{\text{clo}}$. In this paper, we choose to use (22) for the convenience of analysis.

Definition 5 (Price of Anarchy [38]): Price of Anarchy (PoA) is the ratio of system-wide computation costs between the worst (expected) NE and the globally optimal solution in centralized schemes, i.e.,

$$\mathrm{PoA} = \frac{\max\limits_{\boldsymbol{s}^* \in \Psi} \sum_{i \in \mathcal{N}} \theta_i \Gamma_i(\boldsymbol{s}^*)}{\sum_{i \in \mathcal{N}} \theta_i \Gamma_i(\hat{\boldsymbol{s}})}, \tag{24}$$

where θ_i is mobile user i's active probability, Ψ is the set of expected NE of the stochastic game $\mathcal{G}_2$, and $\hat{\boldsymbol{s}}$ denotes the centralized optimal solution that minimizes the system-wide computation cost, i.e., $\hat{\boldsymbol{s}} = \arg\min\limits_{\boldsymbol{s} \in \mathcal{S}} \sum_{i \in \mathcal{N}} \theta_i \Gamma_i(\boldsymbol{s})$, where $\mathcal{S}$ denotes the strategy space of all the users. Notice that the PoA provides a meaningful metric that indicates how good the NE is compared to the centralized optimal solution for minimizing the system cost.

Lemma 3. *For an arbitrary expected NE $\boldsymbol{s}^*$ of the stochastic game $\mathcal{G}_2$, if $s_i^* > 0$, user i's expected transmission rate $\mathbb{E}_\Theta[R_i(\boldsymbol{s}^*, \Theta)]$ for computation offloading is lower bounded by*

$$\bar{R}_i^{\inf} = \mathbb{E}_{g_{i,o}}\left[B\log_2\left(1+\frac{p_i g_{i,o}}{\sigma_0}\right)\right] - B\log_2\left(1+\frac{\sum_{j\in\mathcal{N}\setminus\{i\}} p_j \bar{g}_{j,o}\theta_j}{M\sigma_0}\right). \tag{25}$$

Proof: According to Definition 4 for the expected Nash Equilibrium, $\forall i \in \mathcal{N}$, $\bar{u}_i\left(s_i^*, \boldsymbol{s}_{-i}^*\right) \leq \bar{u}_i\left(s_i, \boldsymbol{s}_{-i}^*\right)$, which along with (8) leads to

$$\mathbb{E}_\Theta\left[I_i\left(s_i^*, \boldsymbol{s}_{-i}^*, \Theta\right)\right] \leq \mathbb{E}_\Theta\left[I_i\left(s_i, \boldsymbol{s}_{-i}^*, \Theta\right)\right], \forall s_i \in \mathcal{M} \subseteq \mathcal{S}_i. \tag{26}$$

By summing up the two sides of (26), we can derive

$$M\mathbb{E}_\Theta\left[I_i\left(s_i^*, \boldsymbol{s}_{-i}^*, \Theta\right)\right] \leq \sum\nolimits_{s_i \in \mathcal{M}} \mathbb{E}_\Theta\left[I_i\left(s_i, \boldsymbol{s}_{-i}^*, \Theta\right)\right]. \tag{27}$$

Obviously, $\mathbb{E}_\Theta\left[I_i\left(s_i, \boldsymbol{s}_{-i}^*, \Theta\right)\right] = \sum_{j\in\mathcal{N}\setminus\{i\}: s_j^* = s_i} p_j \bar{g}_{j,o}\theta_j$. Thus,

$$\begin{aligned}\sum\nolimits_{s_i\in\mathcal{M}} \mathbb{E}_\Theta\left[I_i(s_i, \boldsymbol{s}_{-i}^*, \Theta)\right] &= \sum\nolimits_{s_i\in\mathcal{M}} \sum\nolimits_{j\in\mathcal{N}\setminus\{i\}: s_j^*=s_i} p_j\bar{g}_{j,o}\theta_j \\ &= \sum\nolimits_{s_i\in\mathcal{M}} \sum\nolimits_{j\in\mathcal{N}\setminus\{i\}} p_j\bar{g}_{j,o}\theta_j \ell_{\{s_j^*=s_i\}} \\ &= \sum\nolimits_{j\in\mathcal{N}\setminus\{i\}} p_j\bar{g}_{j,o}\theta_j \sum\nolimits_{s_i\in\mathcal{M}} \ell_{\{s_j^*=s_i\}}.\end{aligned} \tag{28}$$

If $s_j^* = 0$, $\sum_{s_i\in\mathcal{M}} \ell_{\{s_j^*=s_i\}} = 0$; if $s_j^* \in \mathcal{M}$, $\sum_{s_i\in\mathcal{M}} \ell_{\{s_j^*=s_i\}} = 1$. Thus,

$$\sum\nolimits_{s_i\in\mathcal{M}} \mathbb{E}_\Theta\left[I_i\left(s_i, \boldsymbol{s}_{-i}^*, \Theta\right)\right] \leq \sum\nolimits_{j\in\mathcal{N}\setminus\{i\}} p_j\bar{g}_{j,o}\theta_j, \tag{29}$$

which along with (27) yields

$$\mathbb{E}_\Theta\left[I_i\left(s_i^*, \boldsymbol{s}_{-i}^*, \Theta\right)\right] \leq \frac{1}{M}\sum\nolimits_{j\in\mathcal{N}\setminus\{i\}} p_j\bar{g}_{j,o}\theta_j. \tag{30}$$

Besides,

$$\begin{aligned}&\mathbb{E}_\Theta\left[R_i\left(\boldsymbol{s}^*, \Theta\right)\right] = \mathbb{E}_\Theta\left[B\log_2\left(1+\frac{p_i g_{i,o}}{I_i\left(\boldsymbol{s}^*, \Theta\right)+\sigma_0}\right)\right] \\ &= \mathbb{E}_\Theta\left[B\log_2\left(1+\frac{I_i(\boldsymbol{s}^*,\Theta)+p_i g_{i,o}}{\sigma_0}\right)\right] - \mathbb{E}_\Theta\left[B\log_2\left(1+\frac{I_i(\boldsymbol{s}^*,\Theta)}{\sigma_0}\right)\right] \\ &= \mathbb{E}_{g_{i,o}}\left[B\log_2\left(1+\frac{p_i g_{i,o}}{\sigma_0}\right)\right] - \mathbb{E}_{g_{i,o}}\left[B\log_2\left(1+\frac{p_i g_{i,o}}{\sigma_0}\right)\right] \\ &+\mathbb{E}_\Theta\left[B\log_2\left(1+\frac{I_i(\boldsymbol{s}^*,\Theta)+p_i g_{i,o}}{\sigma_0}\right)\right] - \mathbb{E}_\Theta\left[B\log_2\left(1+\frac{I_i(\boldsymbol{s}^*,\Theta)}{\sigma_0}\right)\right].\end{aligned} \tag{31}$$

Notably,

$$\mathbb{E}_{g_{i,o}}\left[B\log_2\left(1+\frac{p_i g_{i,o}}{\sigma_0}\right)\right] \leq \mathbb{E}_\Theta\left[B\log_2\left(1+\frac{I_i\left(\boldsymbol{s}^*,\Theta\right)+p_i g_{i,o}}{\sigma_0}\right)\right]. \tag{32}$$

Moreover, according to Jensen's inequality [41] and the upper bound of expected interference given in (30), we have

$$\begin{aligned}\mathbb{E}_\Theta\left[B\log_2\left(1+\frac{I_i\left(\boldsymbol{s}^*,\Theta\right)}{\sigma_0}\right)\right] &\leq B\log_2\left(1+\frac{\mathbb{E}_\Theta\left[I_i\left(\boldsymbol{s}^*,\Theta\right)\right]}{\sigma_0}\right) \\ &\leq B\log_2\left(1+\frac{\sum_{j\in\mathcal{N}\setminus\{i\}} p_j\bar{g}_{j,o}\theta_j}{M\sigma_0}\right).\end{aligned} \tag{33}$$

Then, (31), (32) and (33) lead to

$$\begin{aligned}\mathbb{E}_\Theta\left[R_i\left(\boldsymbol{s}^*, \Theta\right)\right] \geq &\ \mathbb{E}_{g_{i,o}}\left[B\log_2\left(1+\frac{p_i g_{i,o}}{\sigma_0}\right)\right] \\ &- B\log_2\left(1+\frac{\sum_{j\in\mathcal{N}\setminus\{i\}} p_j\bar{g}_{j,o}\theta_j}{M\sigma_0}\right).\end{aligned} \tag{34}$$

□

Lemma 3 characterizes the lower bound of expected transmission rate of each user for computation offloading at any NE point. According to (25), the lower bound $\bar{R}_i^{\inf}$ increases with the number of available channels M. The reason is that as the number of channels increases, mobile users can avoid mutual interference by choosing different channels for computation offloading. Secondly, when mobile users' active probabilities are lower, the lower bound $\bar{R}_i^{\inf}$ becomes larger, which implies that higher offloading rate can be achieved. With Lemma 3, the following result can be achieved.

Theorem 4. *For the multi-user stochastic computation offloading game $\mathcal{G}_2$, the PoA of the system-wide computation cost satisfies that*

$$1 \leq PoA \leq \frac{\sum\limits_{i=1}^{N} \theta_i \max\left\{V_i^{\mathrm{loc}}, \frac{(\mu_i^E p_i+\mu_i^T)C_i}{\bar{R}_i^{\inf}} + \mu_i^T T_{i,2}^{\mathrm{clo}}\right\}}{\sum\limits_{i=1}^{N} \theta_i \min\left\{V_i^{\mathrm{loc}}, \frac{(\mu_i^E p_i+\mu_i^T)C_i}{\bar{R}_i^{\sup}} + \mu_i^T T_{i,2}^{\mathrm{clo}}\right\}}, \tag{35}$$

where $\bar{R}_i^{\sup} = B\log_2\left(1+\frac{p_i\bar{g}_{i,o}}{\sigma_0}\right)$.

Proof: 1) Let $\boldsymbol{s}^* \in \Psi$ be an arbitrary expected NE of the game $\mathcal{G}_2$. If $s_i^* = 0$, user i chooses local computing with the cost V_i^{loc}; if $s_i^* > 0$, user i chooses cloud computing with the cost $\bar{V}_i^{\mathrm{clo}}(\boldsymbol{s}^*)$. Thus, the following result always holds:

$$\Gamma_i\left(\boldsymbol{s}^*\right) \leq \max\left\{V_i^{\mathrm{loc}}, \bar{V}_i^{\mathrm{clo}}\left(\boldsymbol{s}^*\right)\right\}. \tag{36}$$

Besides, according to Lemma 3, we have

$$\bar{V}_i^{\mathrm{clo}}(\boldsymbol{s}^*) = \frac{\left(\mu_i^E p_i+\mu_i^T\right)C_i}{\mathbb{E}_\Theta[R_i\left(\boldsymbol{s}^*,\Theta\right)]} + \mu_i^T T_{i,2}^{\mathrm{clo}} \leq \frac{\left(\mu_i^E p_i+\mu_i^T\right)C_i}{\bar{R}_i^{\inf}} + \mu_i^T T_{i,2}^{\mathrm{clo}}. \tag{37}$$

Therefore,

$$\Gamma_i\left(\boldsymbol{s}^*\right) \leq \max\left\{V_i^{\mathrm{loc}}, \frac{\left(\mu_i^E p_i+\mu_i^T\right)C_i}{\bar{R}_i^{\inf}} + \mu_i^T T_{i,2}^{\mathrm{clo}}\right\}. \tag{38}$$

2) For the centralized optimal solution $\hat{\boldsymbol{s}}$, the expected transmission rate for computation offloading is

$$\begin{aligned}\mathbb{E}_\Theta[R_i\left(\hat{\boldsymbol{s}},\Theta\right)] &= \mathbb{E}_\Theta\left[B\log_2\left(1+\frac{p_i g_{i,o}}{I_i\left(\hat{\boldsymbol{s}},\Theta\right)+\sigma_0}\right)\right] \\ &\leq \mathbb{E}_{g_{i,o}}\left[B\log_2\left(1+\frac{p_i g_{i,o}}{\sigma_0}\right)\right] \leq B\log_2\left(1+\frac{p_i\bar{g}_{i,o}}{\sigma_0}\right),\end{aligned} \tag{39}$$

where the last inequality in (39) is based on Jensen's inequality [41]. Then, we obtain the following result:

$$\bar{V}_i^{\mathrm{clo}}(\hat{\boldsymbol{s}}) = \frac{\left(\mu_i^E p_i+\mu_i^T\right)C_i}{\mathbb{E}_\Theta[R_i\left(\hat{\boldsymbol{s}},\Theta\right)]} + \mu_i^T T_{i,2}^{\mathrm{clo}} \geq \frac{\left(\mu_i^E p_i+\mu_i^T\right)C_i}{\bar{R}_i^{\sup}} + \mu_i^T T_{i,2}^{\mathrm{clo}}. \tag{40}$$

Since user i's computation cost is either V_i^{loc} or $V_i^{\text{clo}}(\hat{s})$, we have

$$\Gamma_i(\hat{s}) \geq \min\left\{V_i^{\text{loc}}, \bar{V}_i^{\text{clo}}(\hat{s})\right\} \geq \min\left\{V_i^{\text{loc}}, \frac{(\mu_i^E p_i + \mu_i^T) C_i}{\bar{R}_i^{\text{sup}}} + \mu_i^T T_{i,2}^{\text{clo}}\right\}, \quad (41)$$

which along with (38) leads to the upper bound of the PoA given in (35).

Besides, since the centralized optimum $\hat{s}$ minimizes the system-wide computation cost, we hence have PoA ≥ 1. Therefore, we complete the proof of Theorem 4. □

According to Theorem 4 and Lemma 3, more available channels and lower active probabilities help increase the lower bound of expected offloading rate, which thus decreases the gap between the worst NE and the centralized optimal solution.

5.2 Metric II: Beneficial Cloud Computing Users

For an arbitrary NE s^*, let us denote the number of users who benefit from cloud computing by $N^c(s^*)$. Next, we will analyze the bound of $N^c(s^*)$.

Letting $Z_{\max} \triangleq \max_{i\in\mathcal{N}}\{p_i\bar{g}_{i,o}\}$, $Z_{\min} \triangleq \min_{i\in\mathcal{N}}\{p_i\bar{g}_{i,o}\}$, $\bar{Q}_{\max} \triangleq \max_{i\in\mathcal{N}}\{\bar{Q}_i\}$, $\bar{Q}_{\min} \triangleq \min_{i\in\mathcal{N}}\{\bar{Q}_i\}$, $\theta_{\max} \triangleq \max_{i\in\mathcal{N}}\{\theta_i\}$, and $\theta_{\min} \triangleq \min_{i\in\mathcal{N}}\{\theta_i\}$, we can derive the following theorem.

Theorem 5. *Suppose $0 < N^c(s^*) < N$, then for the multi-user dynamic computation offloading game, the total number of users who benefit from cloud computing at any NE point satisfies:*

$$\frac{M\bar{Q}_{\min}}{Z_{\max}\theta_{\max}} \leq N^c(s^*) \leq M\left(\frac{\bar{Q}_{\max}}{Z_{\min}\theta_{\min}} + 1\right). \quad (42)$$

Proof: 1) Since $N^c(s^*) < N$, there exists at least one user k that chooses the local computing manner, i.e., $s_k^* = 0$. Since s^* is an NE, we know that the user cannot reduce its payoff by choosing computation offloading via any channel $m \in \mathcal{M}$. According to (8), we have $\mathbb{E}_\Theta[I_k(s^*, \Theta)] = \sum_{j\in\mathcal{N}\setminus\{k\}} p_j\bar{g}_{j,o}\theta_j\ell_{\{s_k^*=m\}} \geq \bar{Q}_k$, $\forall m \in \mathcal{M}$. Then, let $N_m^c(s^*) = \sum_{i=1}^N \ell_{\{s_i^*=m\}}$ denote the number of users on channel m, and we have

$$N_m^c(s^*) Z_{\max}\theta_{\max} \geq \sum_{j\in\mathcal{N}\setminus\{k\}} p_j\bar{g}_{j,o}\theta_j\ell_{\{s_k^*=m\}} \geq \bar{Q}_k \geq \bar{Q}_{\min}. \quad (43)$$

Thus, $N_m^c(s^*) \geq \frac{\bar{Q}_{\min}}{Z_{\max}\theta_{\max}}$, $\forall m \in \mathcal{M}$. Then, we can obtain

$$N^c(s^*) = \sum_{m=1}^M N_m^c(s^*) \geq \frac{M\bar{Q}_{\min}}{Z_{\max}\theta_{\max}}. \quad (44)$$

2) Since $N^c(s^*) > 0$, there exists at least one user $\tilde{k}$ that chooses the cloud computing manner, i.e., $s_{\tilde{k}}^* > 0$. Without loss of generality, suppose user $\tilde{k}$ is on the channel m, which is occupied by most users, i.e., $N_m^c(s^*) \geq N_{\tilde{m}}^c(s^*)$, $\forall \tilde{m} \in \mathcal{M}$. Since s^* is an NE, we know that the user cannot reduce its payoff by choosing local computation. According to (8), we have $\mathbb{E}_\Theta[I_{\tilde{k}}(s^*, \Theta)] \leq \bar{Q}_{\tilde{k}}$. That is, $\sum_{j\in\mathcal{N}\setminus\{\tilde{k}\}} p_j g_{j,o}\theta_j\ell_{\{s_j^*=m\}} \leq \bar{Q}_{\tilde{k}}$. Then, the following result always holds:

$$(N_m^c(s^*)-1)Z_{\min}\theta_{\min} \leq \sum_{j\in\mathcal{N}\setminus\{\tilde{k}\}} p_j g_{j,o}\theta_j\ell_{\{s_j^*=m\}} \leq \bar{Q}_{\tilde{k}} \leq \bar{Q}_{\max}, \quad (45)$$

which leads to $N_m^c(s^*) \leq \frac{\bar{Q}_{\max}}{Z_{\min}\theta_{\min}} + 1$. Then, we have

$$N^c(s^*) = \sum_{\tilde{m}=1}^M N_{\tilde{m}}^c(s^*) \leq \sum_{\tilde{m}=1}^M N_m^c(s^*) \leq M\left(\frac{\bar{Q}_{\max}}{Z_{\min}\theta_{\min}} + 1\right). \quad (46)$$

Therefore, Theorem 5 is proved. □

Theorem 5 provides a quantitative characterization about how many users can eventually benefit from offloading computation tasks to the mobile cloud, by playing the stochastic game $\mathcal{G}_2$.

Remark 1. *The above analysis indicates that the NE points might enable mobile users to achieve desirable and attractive performance by playing the proposed stochastic game for computation offloading. It is very interesting since mobile users' selfish and competitive behaviors lead to desirable game outcomes. The reasons can be explained as follows. If too many mobile users are using the same wireless channel to offload their computation tasks to the cloud, they may experience low offloading rates. In order to reduce the computation cost, some mobile users will definitely choose other wireless channels for offloading or compute the task locally by itself. Consequently, it leads to balanced occupation of wireless channels for computation offloading, which is beneficial for the whole system.*

6 Multi-Agent Stochastic Learning under Dynamic Environment

Although the NE points exhibit desirable and attractive performance, it is challenging for mobile users to reach the NE in a distributed manner and under dynamic environment. Most existing game-theoretic algorithms (e.g., best (or better) response [24], spatial adaptive play [26]) update users' strategies based on their received instantaneous utility/payoff. However, due to the dynamics of users' activeness and wireless channels in our computation offloading problem, each user might receive different utilities/payoffs in different time slots, even if it chooses to use the same strategy. Thus, the existing game-theoretic algorithms may never reach NE. This motivates us to incorporate the idea of stochastic learning [28], [29], [30] into the design of an efficient yet distributed algorithm under dynamic environment in order to reach the NE of our proposed stochastic game $\mathcal{G}_2$.

6.1 Proposed Multi-Agent Stochastic Learning Algorithm

The details are shown in the Multi-Agent Stochastic Learning Algorithm (i.e., referred as MASL-Algorithm). Specifically, each mobile user acts as a learning automaton that independently and automatically selects its offloading strategy according to a probability vector over the strategy space, and updates the probability vector based on the action-reward received from the dynamic environment. For the sake of clear presentation, we denote the strategy selection probability vector for an arbitrary user i as $\mathbf{w}_i = (w_{i0}, w_{i1}, \ldots, w_{iM})$, where w_{i0} denotes the probability to select the strategy of local computing, w_{im} ($m \in \mathcal{M}$) denotes the probability to select offloading the computation task to the mobile cloud via wireless channel m.

MASL-Algorithm: To reach the NE of the stochastic game $\mathcal{G}_2$

Initialization: At the initial time $t = 0$, each mobile user $i \in \mathcal{N}$ sets its strategy selection probability vector as a uniform distribution, i.e., $\mathbf{w}_i^t = \left(\frac{1}{M+1}, \ldots, \frac{1}{M+1}\right)$.

Loop for $t = 0, 1, 2, \ldots$

1) **Updating computation offloading strategy:** In the t-th time slot, each active user $i \in \mathcal{A}^t$, selects an offloading strategy s_i^t according to its current strategy selection probability vector $\mathbf{w}_i^t$. The inactive users $\mathcal{N}\setminus\mathcal{A}^t$ keep silent and take no action.
2) **Measuring instantaneous payoff**[8]**:** Each active user evaluates its respective payoff u_i^t according to (7), namely, active user i evaluates its received interference I_i^t if $s_i^t > 0$; otherwise, active user i directly computes Q_i^t (which is given in Lemma

8. As we discussed before, the received payoff u_i^t depends not only on other users' activeness, but also on the current channel condition. Specifically, other users' activeness impacts its received interference I_i^t, while the current channel condition impacts both I_i^t and Q_i^t.

1). Besides, the inactive users $\mathcal{N}\backslash\mathcal{A}^t$ keep silent and take no action.

3) **Updating strategy selection probability:** Each active user updates its strategy selection probability vector for the next time slot according to the following rule:

$$\mathbf{w}_i^{t+1} = \mathbf{w}_i^t + br_i^t\left(\mathbf{e}_{s_i^t} - \mathbf{w}_i^t\right), \tag{47}$$

where $0 < b < 1$ is the learning step-size, $\mathbf{e}_{s_i^t}$ is an $(M+1)$-dimensional unit vector with the s_i^t-th element being one, and r_i^t is the received action-reward defined by $r_i^t = 1 - \gamma_i u_i^t$. (The computation offloading strategy with less cost is given larger action-reward. Here γ_i is a scaling factor, and we require $\gamma_i \le \frac{1}{\max_t\{u_i^t\}}$ to guarantee the action-reward r_i^t positive.) Besides, the inactive mobile users $\mathcal{N}\backslash\mathcal{A}^t$ keep their strategy selection probability vectors unchanged, i.e.,

$$\mathbf{w}_i^{t+1} = \mathbf{w}_i^t. \tag{48}$$

End loop until all users do not adjust their respective offloading strategies.

As shown above, the proposed MASL-Algorithm is operated in an iterative manner. Within each round of iteration, each active user independently selects its offloading strategy based on a probability vector over the strategy space, and receives an action-reward from the dynamic environment. The computation offloading strategy with less cost is given larger action-reward, and the strategy with larger reward value will be assigned with larger probability. By continuously interacting with the random environment, each mobile user will finally choose its optimal offloading strategy with probability one. We emphasize that during the operation of MASL-Algorithm, each mobile user operates entirely based on its own strategies and the consequently received reward, without requiring any knowledge from other users and any prior knowledge of probability space $(\Omega, \mathcal{H}, \mathbb{P})$ of the dynamic environment. Therefore, the proposed MASL-Algorithm is *fully distributed* which makes itself attractive for a practical implementation.

Notice that due to the dynamics of users' activeness and wireless channels, user i might receive different action-rewards in different time slots, even if it chooses to use the same strategy. This imposes the key challenge to establish the convergence of MASL-Algorithm. Although there exist several previous studies [28], [29], [30] investigating the convergence of some stochastic learning algorithms, our proposed MASL-Algorithm differs from those algorithms in terms of taking into account the dynamics of both users' activeness and wireless channels. Moreover, the definition of payoff function is application-dependent, and different payoff functions (adopted by even the same learning mechanism) will lead to different learning solutions [31]. Therefore, the previous analyses are not applicable to our case in this study, which motivates us to perform a deep analysis about the convergence property of the MASL-Algorithm in the next subsection.

6.2 Convergence Properties of MASL-Algorithm

We first re-write the updating rule in Step 3 of the proposed algorithm as follows:

$$\mathbf{w}_i^{t+1} = \mathbf{w}_i^t + ba_i^t r_i^t\left(\mathbf{e}_{s_i^t} - \mathbf{w}_i^t\right), \quad \forall i \in \mathcal{N}, \tag{49}$$

where a_i^t denotes the activeness of user i in the t-th time slot. Let $\mathbf{W}^t = \left(\mathbf{w}_1^t, \ldots, \mathbf{w}_N^t\right)^T$ denote the strategy selection probability vector of all the users, and thus we can express the evolution of the strategy selection probability vector of the game $\mathcal{G}_2$ as follows:

$$\mathbf{W}^{t+1} = \mathbf{W}^t + bf\left(\mathbf{W}^t, \boldsymbol{a}^t, \boldsymbol{s}^t, \boldsymbol{r}^t\right), \tag{50}$$

where $\boldsymbol{a}^t = (a_1^t, \ldots, a_N^t)$, $\boldsymbol{s}^t = (s_1^t, \ldots, s_N^t)$, $\boldsymbol{r}^t = (r_1^t, \ldots, r_N^t)$, and $f(\cdot)$ represents the updating rule specified by (49). Then, according to Theorem 3.1 in [28], we can derive the following lemma.

Lemma 4. *With a sufficiently small step-size b, i.e., $b \to 0$, the sequence $\{\mathbf{W}^t\}$ will converge weakly to the solution of the following ordinary differential equation (ODE):*

$$\frac{d\mathbf{W}}{dt} = h(\mathbf{W}), \tag{51}$$

with the initial state $\mathbf{W}^0 = \left[\frac{1}{M+1}\right]_{N\times(M+1)}$, and $h(\mathbf{W}) = \mathbb{E}_\Theta\left[f\left(\mathbf{W}^t, \boldsymbol{a}^t, \boldsymbol{s}^t, \boldsymbol{r}^t, \Theta\right)|\mathbf{W}^t = \mathbf{W}\right]$.

Lemma 5. *With a sufficiently small step-size b, our proposed MASL-Algorithm converges to a stable stationary point of the ODE given in (51).*

Proof: Let $\bar{r}_i(\boldsymbol{s}) = \mathbb{E}_\Theta[r_i(\boldsymbol{s}, \Theta)]$ denote user i's expected reward function under strategy profile $\boldsymbol{s}$, and let $X_i(m, \mathbf{W}_{-i})$ denote user i's probabilistic reward function when it adopts pure strategy m and other users adopt probability vector (for strategy selection) $\mathbf{W}_{-i} = (\mathbf{w}_1, \ldots \mathbf{w}_{i-1}, \mathbf{w}_{i+1}, \ldots, \mathbf{w}_N)$. Specifically, define

$$X_i(m, \mathbf{W}_{-i}) = \bar{r}_i(m, \mathbf{W}_{-i}) = \sum_{\boldsymbol{s}_{-i}\in\mathcal{S}_{-i}} \bar{r}_i(m, \boldsymbol{s}_{-i}) \prod_{j\neq i} w_{j,s_j}, \tag{52}$$

where $\mathcal{S}_{-i}$ denotes the strategy space of all the users excluding i, w_{j,s_j} is the probability of user j to choose pure strategy s_j. In addition, define the probabilistic potential function

$$Y(\mathbf{W}) = \bar{\Phi}(\mathbf{W}) = \sum_{\boldsymbol{s}\in\mathcal{S}} \bar{\Phi}(\boldsymbol{s}) \prod_{i\in\mathcal{N}} w_{i,s_i}, \tag{53}$$

and

$$Y_i(m, \mathbf{W}_{-i}) = \frac{\partial Y(\mathbf{W})}{\partial w_{i,m}} = \sum_{\boldsymbol{s}_{-i}\in\mathcal{S}_{-i}} \bar{\Phi}(m, \boldsymbol{s}_{-i}) \prod_{j\neq i} w_{j,s_j}, \tag{54}$$

where $\bar{\Phi}$ is the expected potential function defined in (20), $\mathcal{S}$ denotes the strategy space of all the users.

According to Lemma 4, we have

$$\begin{aligned}\frac{dw_{i,m}}{dt} &= \theta_i\left(w_{i,m}(1-w_{i,m})\bar{r}_i(m,\mathbf{W}_{-i}) + \sum_{m'\neq m} w_{i,m'}(-w_{i,m})\,\bar{r}_i(m',\mathbf{W}_{-i})\right) \\ &= \theta_i w_{i,m}\left(\bar{r}_i(m, \mathbf{W}_{-i}) - \sum_{m'\in\mathcal{S}_i} w_{i,m'}\bar{r}_i(m', \mathbf{W}_{-i})\right) \\ &= \theta_i w_{i,m}\left(X_i(m, \mathbf{W}_{-i}) - \sum_{m'\in\mathcal{S}_i} w_{i,m'} X_i(m', \mathbf{W}_{-i})\right) \\ &= \theta_i w_{i,m} \sum_{m'\in\mathcal{S}_i} w_{i,m'}\left(X_i(m, \mathbf{W}_{-i}) - X_i(m', \mathbf{W}_{-i})\right).\end{aligned} \tag{55}$$

Then,

$$\begin{aligned}\frac{dY(\mathbf{W})}{dt} &= \sum_{i,m} \frac{\partial Y(\mathbf{W})}{\partial w_{i,m}} \frac{dw_{i,m}}{dt} \\ &= \sum_{i,m} Y_i(m,\mathbf{W}_{-i})\,\theta_i w_{i,m} \sum_{m'\in\mathcal{S}_i} w_{i,m'}\left(X_i(m,\mathbf{W}_{-i}) - X_i(m',\mathbf{W}_{-i})\right) \\ &= \sum_{i,m,m'} \theta_i w_{i,m} w_{i,m'} Y_i(m, \mathbf{W}_{-i})\left(X_i(m, \mathbf{W}_{-i}) - X_i(m', \mathbf{W}_{-i})\right).\end{aligned} \tag{56}$$

Notably,

$$\sum_{i,m,m'} \theta_i w_{i,m} w_{i,m'} Y_i(m, \mathbf{W}_{-i}) \left(X_i(m, \mathbf{W}_{-i}) - X_i(m', \mathbf{W}_{-i})\right) = \sum_{i,m,m'} \theta_i w_{i,m'} w_{i,m} Y_i(m', \mathbf{W}_{-i}) \left(X_i(m', \mathbf{W}_{-i}) - X_i(m, \mathbf{W}_{-i})\right). \tag{57}$$

Therefore, we can obtain the following result:

$$\frac{dY(\mathbf{W})}{dt} = \frac{1}{2} \sum_{i,m,m'} \theta_i w_{i,m} w_{i,m'} \left(Y_i(m, \mathbf{W}_{-i}) - Y_i(m', \mathbf{W}_{-i})\right) \times \left(X_i(m, \mathbf{W}_{-i}) - X_i(m', \mathbf{W}_{-i})\right). \tag{58}$$

As given in Step 3 of the proposed algorithm, $\forall t$, $r_i^t = 1 - \gamma u_i^t$, thus the expected reward $\bar{r}_i = 1 - \gamma \bar{u}_i$. Then, (21) yields

$$\bar{\Phi}(s_i', s_{-i}) - \bar{\Phi}(s_i, s_{-i}) = \frac{p_i \bar{g}_{i,o}}{\gamma_i} \left(\bar{r}_i(s_i, s_{-i}) - \bar{r}_i(s_i', s_{-i})\right). \tag{59}$$

By using Eqs. (52), (54), and (59), we can derive that $\forall m, m' \in \mathcal{S}_i$,

$$Y_i(m, \mathbf{W}_{-i}) - Y_i(m', \mathbf{W}_{-i}) = \frac{p_i \bar{g}_{i,o}}{\gamma_i} \left(X_i(m', \mathbf{W}_{-i}) - X_i(m, \mathbf{W}_{-i})\right), \tag{60}$$

which, together with (58), yields

$$\frac{dY(\mathbf{W})}{dt} = -\frac{1}{2} \sum_{i,m,m'} \theta_i w_{i,m} w_{i,m'} \frac{p_i \bar{g}_{i,o}}{\gamma_i} \left(X_i(m, \mathbf{W}_{-i}) - X_i(m', \mathbf{W}_{-i})\right)^2 \leq 0. \tag{61}$$

(61) indicates that $Y(\mathbf{W})$ monotonously decreases when the algorithm iterates. Moreover, since $Y(\mathbf{W})$ is lower bounded by $Y(\mathbf{W}) \geq 0$, we know $Y(\mathbf{W})$ will converge to a stationary point when $\frac{dY(\mathbf{W})}{dt} = 0$, and

$$\frac{dY(\mathbf{W})}{dt} = 0 \Rightarrow w_{i,m} w_{i,m'} \left(X_i(m, \mathbf{W}_{-i}) - X_i(m', \mathbf{W}_{-i})\right) = 0. \tag{62}$$

Then, according to (55), we have $\frac{dw_{i,m}}{dt} = 0, \forall i, m$, and thus $\frac{d\mathbf{W}}{dt} = 0$. Hence, $\mathbf{W}$ converges to a stationary point of ODE (51). This completes the proof of Lemma 5. □

It has been proved by Theorem 3.2 in [28] that all pure-strategy NE of $\mathcal{G}_2$ coincide with the stable stationary points of the ODE given in (51). Thus, based on Lemma 5, we can derive the following theorem.

Theorem 6. *With a sufficiently small step-size b, our proposed MASL-Algorithm converges to a pure-strategy NE point of $\mathcal{G}_2$.*

Moreover, the convergence rate of the MASL-Algorithm can be characterized as follows:

Theorem 7. *The average convergence rate of the proposed MASL-Algorithm is given by:*

$$\rho^{ave} = \sqrt{\rho^0 \rho^\infty}, \tag{63}$$

with

$$\rho^0 = 1 - \frac{b \sum_{i \in \mathcal{N}} \sum_{m \in \mathcal{S}_i} \sum_{m' \in \mathcal{S}_i} \sum_{\boldsymbol{s}_{-i} \in \mathcal{S}_{-i}} \theta_i \gamma_i p_i \bar{g}_{i,o} \left(\bar{u}_i(m', \boldsymbol{s}_{-i}) - \bar{u}_i(m, \boldsymbol{s}_{-i})\right)^2}{2(M+1)^N \sum_{\boldsymbol{s} \in \mathcal{S}} \left(\bar{\Phi}(\boldsymbol{s}) - \bar{\Phi}(\boldsymbol{s}^*)\right)}, \tag{64}$$

and

$$\rho^\infty = 1 - \frac{b \sum_{i \in \mathcal{N}} \sum_{s_i \in \mathcal{S}_i} \theta_i \gamma_i p_i \bar{g}_{i,o} \left(\bar{u}_i(s_i, \boldsymbol{s}_{-i}^*) - \bar{u}_i(\boldsymbol{s}^*)\right)^2}{\sum_{i \in \mathcal{N}} \sum_{s_i \in \mathcal{S}_i} \left(\bar{\Phi}(s_i, \boldsymbol{s}_{-i}^*) - \bar{\Phi}(\boldsymbol{s}^*)\right)}, \tag{65}$$

where $\mathcal{S}_{-i}$ denotes the strategy space of all the users excluding i, $\mathcal{S}$ denotes the strategy space of all the users, and $\boldsymbol{s}^*$ is a pure-strategy NE.

Proof: As shown in (61), the probabilistic potential function $Y(\mathbf{W}) = \bar{\Phi}(\mathbf{W})$ monotonously converges. Specifically, at iteration index t, the corresponding convergence rate is given by:

$$\rho^t = \frac{\bar{\Phi}(\mathbf{W}^{t+1}) - \bar{\Phi}(\mathbf{W}^*)}{\bar{\Phi}(\mathbf{W}^t) - \bar{\Phi}(\mathbf{W}^*)} = 1 + \frac{Y(\mathbf{W}^{t+1}) - Y(\mathbf{W}^t)}{Y(\mathbf{W}^t) - Y(\mathbf{W}^*)}, \tag{66}$$

where $\mathbf{W}^*$ is a stationary point of the ODE, i.e., an NE. As stated in [39], ρ^t indicates how close $\bar{\Phi}(\mathbf{W}^{t+1})$ is to $\bar{\Phi}(\mathbf{W}^*)$, compared with $\bar{\Phi}(\mathbf{W}^t)$.

$$\begin{aligned} Y(\mathbf{W}^{t+1}) - Y(\mathbf{W}^t) &\approx \sum_{i,m} \frac{\partial Y(\mathbf{W})}{\partial w_{i,m}} |_{\mathbf{W}^t} \Delta w_{i,m} \\ &= \sum_{i,m} Y_i(m, \mathbf{W}_{-i}^t) \theta_i b w_{i,m} \sum_{m' \in \mathcal{S}_i} w_{i,m'} \left(X_i(m, \mathbf{W}_{-i}^t) - X_i(m', \mathbf{W}_{-i}^t)\right) \\ &= b \sum_{i,m,m'} \theta_i w_{i,m}^t w_{i,m'}^t Y_i(m, \mathbf{W}_{-i}^t) \left(X_i(m, \mathbf{W}_{-i}^t) - X_i(m', \mathbf{W}_{-i}^t)\right) \\ &= -\frac{b}{2} \sum_{i,m,m'} \theta_i w_{i,m}^t w_{i,m'}^t \frac{p_i \bar{g}_{i,o}}{\gamma_i} \left(X_i(m, \mathbf{W}_{-i}^t) - X_i(m', \mathbf{W}_{-i}^t)\right)^2. \end{aligned} \tag{67}$$

Besides, [28], [40] show that only the pure-strategy NE is stable, thus we only study the convergence to a pure-strategy NE $\boldsymbol{s}^*$. Obviously, $Y(\mathbf{W}^*)$ can be equivalently written as $\bar{\Phi}(\mathbf{s}^*)$. Therefore,

$$\rho^t = 1 - \frac{\frac{b}{2} \sum_{i,m,m'} \theta_i w_{i,m}^t w_{i,m'}^t \frac{p_i \bar{g}_{i,o}}{\gamma_i} \left(X_i(m, \mathbf{W}_{-i}^t) - X_i(m', \mathbf{W}_{-i}^t)\right)^2}{\sum_{\boldsymbol{s} \in \mathcal{S}} \bar{\Phi}(\boldsymbol{s}) \prod_{i \in \mathcal{N}} w_{i,s_i}^t - \bar{\Phi}(\boldsymbol{s}^*)}, \tag{68}$$

which along with (52) yields

$$\begin{aligned} \rho^t &= 1 - \frac{\frac{b}{2} \sum_{i,m,m'} \sum_{\boldsymbol{s}_{-i} \in \mathcal{S}_{-i}} \theta_i \frac{p_i \bar{g}_{i,o}}{\gamma_i} \left(\bar{r}_i(m, \boldsymbol{s}_{-i}) - \bar{r}_i(m', \boldsymbol{s}_{-i})\right)^2 w_{i,m}^t w_{i,m'}^t \left(\prod_{j \neq i} w_{j,s_j}^t\right)^2}{\sum_{\boldsymbol{s} \in \mathcal{S}} \bar{\Phi}(\boldsymbol{s}) \prod_{i \in \mathcal{N}} w_{i,s_i}^t - \bar{\Phi}(\boldsymbol{s}^*)} \\ &= 1 - \frac{\frac{b}{2} \sum_{i,m,m'} \sum_{\boldsymbol{s}_{-i} \in \mathcal{S}_{-i}} \theta_i \gamma_i p_i \bar{g}_{i,o} \left(\bar{u}_i(m', \boldsymbol{s}_{-i}) - \bar{u}_i(m, \boldsymbol{s}_{-i})\right)^2 \prod_{i \in \mathcal{N}: s_i = m} w_{i,s_i}^t \prod_{i \in \mathcal{N}: s_i = m'} w_{i,s_i}^t}{\sum_{\boldsymbol{s} \in \mathcal{S}} \bar{\Phi}(\boldsymbol{s}) \prod_{i \in \mathcal{N}} w_{i,s_i}^t - \bar{\Phi}(\boldsymbol{s}^*)}. \end{aligned} \tag{69}$$

At $t = 0$, $w_{i,s_i}^0 = \frac{1}{M+1}, \forall i \in \mathcal{N}, \forall s_i \in \mathcal{S}_i$, thus, we can derive ρ^0 as (64). At $t = \infty$, let the probability in NE $\prod_{i \in \mathcal{N}} w_{i,s_i^*}^\infty = 1 - \varepsilon$, the probability for only one user deviating NE $w_{j,s_j}^\infty \prod_{i \in \mathcal{N} \setminus \{j\}} w_{i,s_i^*}^\infty = \frac{\varepsilon}{NM}$, $s_j \neq s_j^*$, and the probability for more than one user deviating NE to be 0. As $\varepsilon \to 0$, we obtain ρ^∞ as (65). Then, according to the definition in [39], the average rate of convergence can be computed by $\rho^{\text{ave}} = \sqrt{\rho^0 \rho^\infty}$. Notably, the time taken for $\bar{\Phi}(\mathbf{W}^0) - \bar{\Phi}(\mathbf{W}^*)$ to decrease L times its value is $T^{\text{ave}} = \frac{\log L}{\log \rho^{\text{ave}}}$. □

Remark 2. *(The impact of step-size b) It is noted that the convergence rate is higher when ρ^{ave} is smaller [39]. Notably, ρ^{ave} decreases linearly with the value of b. Thus, when b is larger, the convergence gets faster. However, the approximation of the iterative process to the ODE requires a sufficiently small b, as shown in Lemma 4. [40] has characterized the accuracy of approximation as*

$$\mathbb{E}\left[\mathbf{W}^t - \tilde{\mathbf{W}}^{bt}\right] = O\left(\sqrt{b}\right), \tag{70}$$

where $\mathbf{W}^t$ is the iterative process of algorithm, and $\tilde{\mathbf{W}}^{bt}$ represents the value of the trajectory of ODE at time bt. Therefore, there is a tradeoff between the convergence rate and the accuracy of approximation to NE

TABLE 2
Parameters Setting

Coverage of AP	50 m	Data size C_i	5000 KB
Number of mobile users N	[20, 45]	Number of CPU cycles for local computing D_i^{loc}	1000 Megacycles
User's active probability θ_i	(0, 1]	Number of CPU cycles for cloud computing D_i^{clo}	1200 Megacycles
Number of channels M	[4, 14]	Local computational capability F_i^{loc}	Randomly set from {0.5, 0.8, 1.0} GHz
Bandwidth of channel B	5 MHz	Cloud computational capability F_i^{clo}	12 GHz
Transmit power p_i	100 mW	Weight of computational energy μ_i^E	Randomly set from {0, 0.5, 1.0}
Pass loss exponent α	4	Weight of computational time μ_i^T	$1-\mu_i^E$
Background noise σ_0	-100 dBm	Computing energy efficiency $1/\eta_i$	Randomly set from {400, 500, 600} Megacycles/J

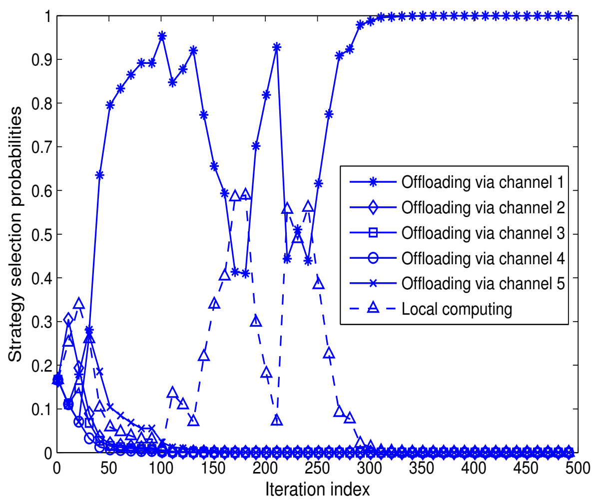


Fig. 3. Convergence of strategy selection probabilities.

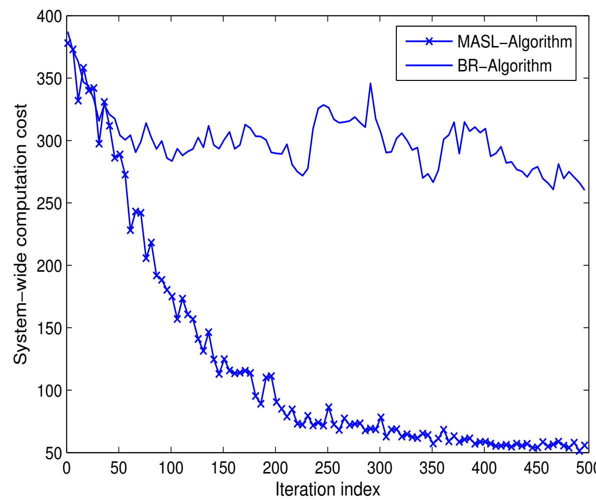


Fig. 4. Convergence of system-wide computation cost.

of the stochastic game, and the parameter setting of b is application-dependent in practice. Figure 5 in the next section verifies the above discussions.

7 Simulation Results

In this section, we conduct simulations to validate the proposed MASL-Algorithm and its performance under dynamic environment. We set up a scenario network where a group of mobile users are randomly scattered in the coverage of the AP. For each user i, we randomly set its active probability θ_i according to a uniform distribution within (0,1]. The time-varying channels follow Rayleigh fading, where the random fading coefficient β is exponentially distributed with unit-mean. Table 2 summarizes the key parameters used in simulations, which are set similar to [10], [11]. If not otherwise specified, the default setting for the number of users is 30, and that for the number of channels is 5. Moreover, in the proposed MASL-Algorithm, the default value of the step-size b is 0.1, and that of the scaling factor γ_i is 10^5 (the impact will be discussed later on).

7.1 Convergence Analysis

To evaluate convergence of the proposed MASL-Algorithm, we plot the strategy selection probabilities of one arbitrarily selected user in Figure 3. At the very beginning, this target user randomly selects its computation offloading strategy according to a uniform distribution. As the MASL-Algorithm operates, this target user's strategy selection probabilities keep on updating and finally converge after around 300 iterations[9]. Specifically, after convergence, the probability of choosing offloading computation task via channel 1 is equal to 1, while the probabilities for other strategies decrease to 0. This result means that, after convergence, the target user will only choose channel 1 to offload computation task to the mobile cloud.

Figure 4 plots the dynamics of system-wide computation cost for our MASL-Algorithm. For comparison, the best-response algorithm (referred as BR-Algorithm) in [11] is plotted, which also runs in an iterative manner. At each iteration, the BR-Algorithm allows the user who received the update-permission message to select its optimal strategy based on its *instantaneous* computation cost, while other users keep their strategies unchanged. We can see that the MASL-Algorithm can greatly reduce the computation cost to its minimum (i.e., NE) after convergence, while the BS-Algorithm yields a fluctuating computational cost which is much greater than that of the MASL-Algorithm. This is because the BR-Algorithm always conducts a myopic play based on the instantaneous computation cost, while the environment is dynamically varying.

We then analyze the computational complexity of the proposed algorithm. Since most operations only involve some basic arithmetical calculations, the dominating part is the updating of the strategy selection probability in Step 3, which involves the operations of 2 vector-vector sums, 1 scalar-vector product, and 1 scalar-scalar product. Thus, the proposed MASL-Algorithm runs

9. Since the typical length of a time slot in wireless systems is at the time scale of microseconds (e.g., 70 microseconds for a time slot in LTE system [32]), the time used by the computation offloading decision process is actually very short (e.g., 21 milliseconds for 300 iterations in LTE system). Such a short duration is negligible, compared with the computation execution process, which is typically at the scale of seconds (e.g., the execution time for mobile gaming applications is typically several hundred milliseconds [37]).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2018.2847337, IEEE Transactions on Mobile Computing

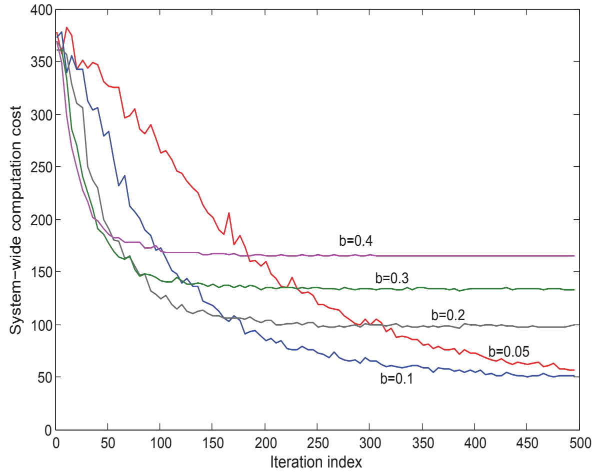


Fig. 5. Impact of different values of step-size b.

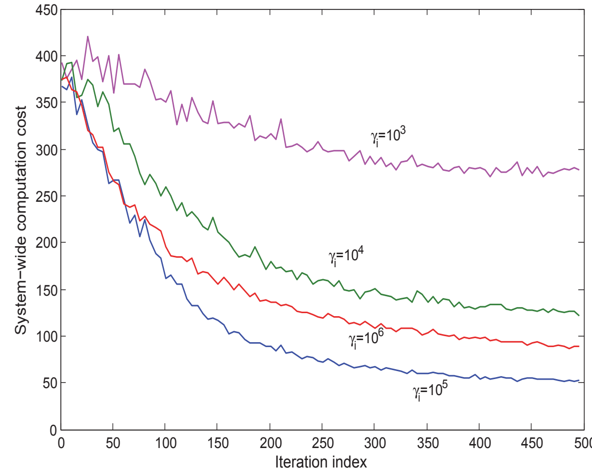


Fig. 6. Impact of different values of scaling factor γ_i.

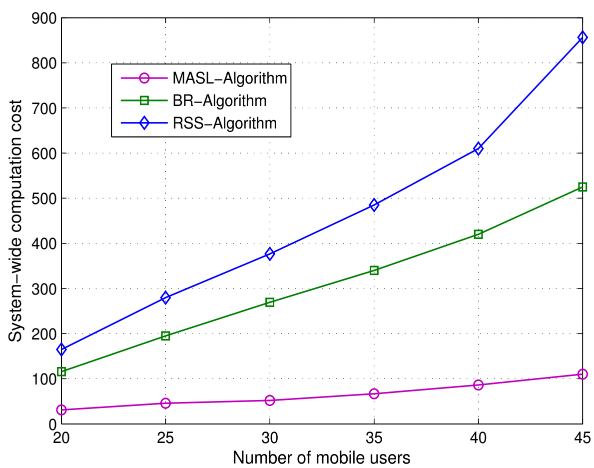


Fig. 7. Performance comparison in terms of system cost for different numbers of mobile users.

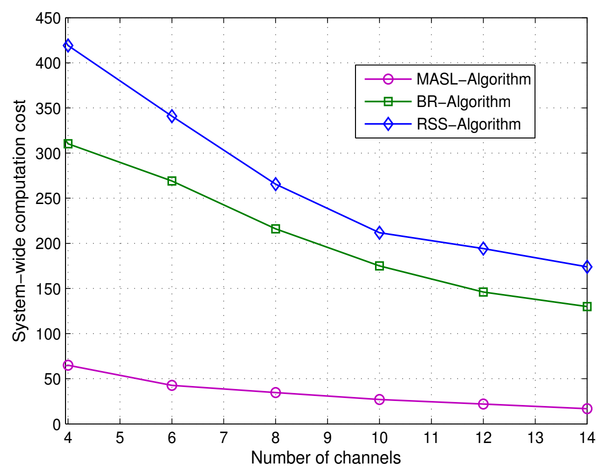


Fig. 8. Performance comparison in terms of system cost for different numbers of channels.

in a low complexity of $\mathcal{O}(3M+1)$. In contrast, [11] has shown that the computational complexity of the BR-Algorithm is $\mathcal{O}(M \log M)$.

Figure 5 shows the impact of the step-size on the convergence speed of the proposed MASL-Algorithm. We vary the step-size b to be 0.05, 0.1, 0.2, 0.3, and 0.4, respectively. Figure 5 shows that as the step-size increases, the algorithm can speed up its convergence, but obtains an inferior solution (not NE). As shown in Remark 2 (Section VI), there is a tradeoff between the convergence speed and the accuracy of convergence to NE, which are both impacted by the step-size b. For our case, it is preferable to set the step-size as 0.1, which can lead the algorithm converge to an NE within about 350 iterations. It is noted setting the step-size as 0.05 can also get the NE, but the convergence speed is very slow (more than 500 iterations).

Moreover, Figure 6 shows the impact of the scaling factor γ_i on the performance of the proposed MASL-Algorithm. We vary the parameter γ_i to be 10^3, 10^4, 10^5, and 10^6, respectively. Figure 6 shows that when $\gamma_i = 10^5$, the algorithm achieves the best performance in terms of reducing the system-wide computation cost. Larger γ_i setting could enhance users' response to the computation cost, and thus lead users' sensitive strategy adjustment towards the optimal one. That is why we observe from the figure that 10^5 is better than 10^3 and 10^4 for the setting of γ_i. However, γ_i cannot be too large in order to guarantee the action-reward r_i^t positive. As a result, the performance of the algorithm gets worse when $\gamma_i = 10^6$.

7.2 Performance Evaluation

We further evaluate the performance of the proposed MASL-Algorithm, in comparison with the performance of the BR-Algorithm [11] and the random strategy selection algorithm (referred as RSS-Algorithm). Specifically, in the RSS-Algorithm, each mobile user randomly selects a strategy in each time slot. The following presented results are obtained by simulating 500 independent trials and then taking the average value.

Figure 7 shows the performance of our proposed MASL-Algorithm versus different numbers of mobile users. Figure 7 shows that the MASL-Algorithm always consumes a significantly less total cost than the BR-Algorithm and the RSS-Algorithm. In addition, the consumed total cost increases as the number of mobile users increases, which is consistent with the intuition. Figure 8 further shows the advantage of using MASL-Algorithm by varying different numbers of channels. Figure 8 again shows that our proposed MASL-Algorithm always outperforms the BR-Algorithm and the RSS-Algorithm.

Figures 9 and 10 evaluate our proposed MASL-Algorithm in terms of the number of mobile users who benefit from performing cloud computing (notice that the decimals in the results are due

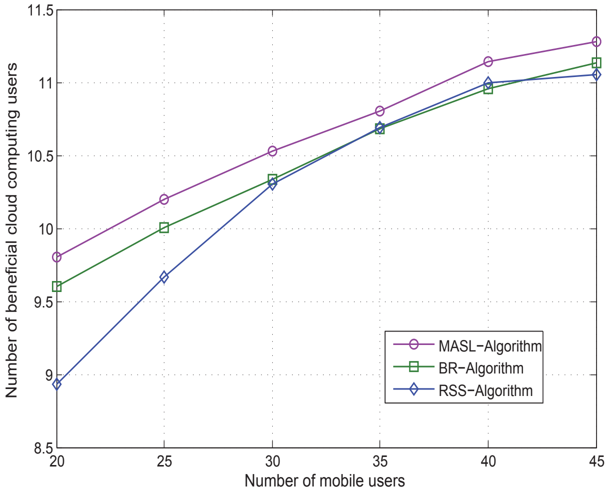


Fig. 9. Performance comparison in terms of beneficial cloud computing users for different numbers of users.

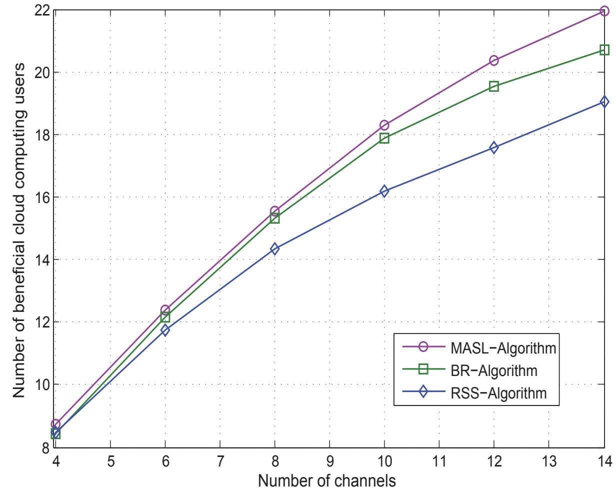


Fig. 10. Performance comparison in terms of beneficial cloud computing users for different numbers of channels.

to the average results). Both figures demonstrate that our proposed MASL-Algorithm achieves better performances than the BR-Algorithm and the RSS-Algorithm. Figure 9 shows that the number of users who benefit from cloud computing increases when the number of users becomes larger. Intuitively, when the number of total users increases, more users may possibly choose cloud computing. However, due to limitation of available channels, the number of beneficial cloud computing users is also limited, since users would generate severe interference to each other, leading to lower offloading rates. Figure 10 shows that the number of beneficial cloud computing users increases when the number of available channels increases.

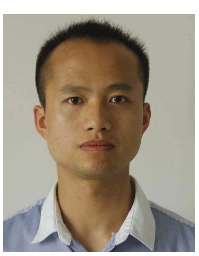
8 Conclusion

In this paper, we have investigated the problem of multi-user computation offloading for mobile cloud computing under the practical dynamic environment. By formulating this problem as a stochastic game, we proved that such a dynamic offloading decision process always leads to a pure-strategy NE. Moreover, we have analyzed the performance bounds of the NE in terms of both the system cost and the number of users who can benefit from cloud computing. To reach the NE, we proposed a fully distributed algorithm (i.e., MASL-Algorithm) with a guaranteed convergence rate under dynamic environment. Simulation results have been presented to validate the effectiveness of our proposed algorithm and show its significant performance advantage. In our future work, we will study the joint optimization of dynamic offloading decision-making and transmit power control, which will be an important and technical challenging problem. Another interesting direction is to investigate the mobile computation offloading from the perspective of economics, and specifically, to consider mobile users' economic expenses for offloading computation tasks to the mobile cloud.

References

[1] J. Zheng, Y. Cai, Y. Wu, and X. Shen, "Stochastic computation offloading game for mobile cloud computing," in *Proc. IEEE/CIC Int. Conf. on Commun. in China (ICCC)*, 2016, pp. 1-6.
[2] Y. Xu and S. Mao, "A survey of mobile cloud computing for rich media applications," *IEEE Wireless Commun.*, vol. 20, no. 3, pp. 46-53, June 2013.
[3] Y. Zhao, S. Zhou, T. Zhao, and Z. Niu, "Energy-efficient task offloading for multiuser mobile cloud computing," in *Proc. IEEE/CIC Int. Conf. on Commun. in China (ICCC)*, 2015, pp. 1-5.
[4] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE Trans. Signal and Information Process. over Netw.*, vol. 1, no. 2, pp. 89-103, June 2015.
[5] Y. Cai, F. R. Yu, and S. Bu, "Cloud computing meets mobile wireless communications in next generation cellular networks," *IEEE Netw.*, vol. 28, no. 6, pp. 54-59, 2014.
[6] Y. Wen, W. Zhang, and H. Luo, "Energy-optimal mobile application execution: Taming resource-poor mobile devices with cloud clones," in *Proc. IEEE INFOCOM*, 2012, pp. 2716-2720.
[7] M. V. Barbera, S. Kosta, A. Mei, and J. Stefa, "To offload or not to offload? the bandwidth and energy costs of mobile cloud computing," in *Proc. IEEE INFOCOM*, 2013, pp. 1285-1293.
[8] J. Zheng, Y. Wu, N. Zhang, H. Zhou, Y. Cai, and X. Shen, "Optimal power control in ultra-dense small cell networks: A game-theoretic approach," *IEEE Trans. Wireless Commun.*, vol. 16, no. 7, pp. 4139-4150, Jul. 2017.
[9] D. Niyato, E. Hossain, and Z. Han, "Dynamics of multiple-seller and multiple-buyer spectrum trading in cognitive radio networks: A game-theoretic modeling approach," *IEEE Trans. Mobile Computing*, vol. 8, no. 8, pp. 1009-1022, Aug. 2009.
[10] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 974-983, Apr. 2015.
[11] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2795-2808, Oct. 2016.
[12] T. Lin, T. Alpcan, K. Hinton, "A game-theoretic analysis of energy efficiency and performance for cloud computing in communication networks," *IEEE Syst. J.*, vol. 11, no. 2, pp. 649-660, Jun. 2017.
[13] V. Cardellini, V. D. N. Personé, V. D. Valerio, F. Facchinei, V. Grassi, F. L. Presti, and V. Piccialli, "A game-theoretic approach to computation offloading in mobile cloud computing," *Math. Program.*, vol. 157, no. 2, pp. 421-449, Jun. 2016.
[14] M. Chen, M. Dong, and B. Liang, "Multi-user mobile cloud offloading game with computing access point," in *Proc. IEEE Cloudnet*, 2016, pp. 64-69.
[15] Y. Wang, S. Meng, Y. Chen, R. Sun, X. Wang, and K. Sun, "Multi-leader multi-follower Stackelberg game based dynamic resource allocation for mobile cloud computing environment," *Wireless Pers. Commun.*, vol. 93, no. 2, pp. 461-480, Mar. 2017.
[16] Y. Liu, C. Xu, Y. Zhan, Z. Liu, J. Guan, and H. Zhang, "Incentive mechanism for computation offloading using edge computing: A Stackelberg game approach," *Comput. Netw.*, vol. 129, no. 2, pp. 399-409, Dec. 2017.
[17] A. Rudenko, P. Reiher, G. J. Popek, and G. H. Kuenning, "Saving portable computer battery power through remote process execution," *Mobile Comput. Commun. Rev.*, vol. 2, no. 1, pp. 19-26, 1998.
[18] C. Xian, Y. Lu, and Z. Li, "Adaptive computation offloading for energy conservation on battery-powered systems," in *Proc. IEEE ICDCS*, 2007, pp. 1-8.
[19] G. Huertacanepa and D. Lee, "An adaptable application offloading scheme based on application behavior," in *Proc. 22nd Int. Conf. Adv. Inf. Netw. Appl.-Workshops*, 2008, pp. 387-392.
[20] J. Kwak, Y. Kim, J. Lee, and S. Chong, "DREAM: Dynamic resource and task allocation for energy minimization in mobile cloud systems," *IEEE J. Sel. Areas Commun.*, vol 33. no. 12, pp. 2510-2523, Dec. 2015.

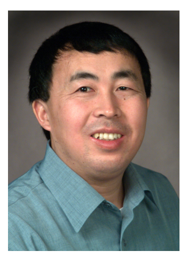
[21] D. Huang, P. Wang, and D. Niyato, "A dynamic offloading algorithm for mobile computing," *IEEE Trans. Wireless Commun.*, vol. 11, no. 6, pp. 1991-1995, Jun. 2012.
[22] H. Wu, D. Huang, and S. Bouzefrane, "Making offloading decisions resistant to network unavailability for mobile cloud collaboration," in *Proc. IEEE Collaboratecom*, 2013, pp. 168-177.
[23] L. Yang, J. Cao, S. Tang, T. Li, and A. T. S. Chan, "A framework for partitioning and execution of data stream applications in mobile cloud computing," in *Proc. IEEE Int. Conf. Cloud Computing*, 2012, pp. 794-802.
[24] D. Monderer and L. S. Shapley, "Potential games," *Games and Economic Behavior*, vol. 14, pp. 124-143, 1996.
[25] J. Marden, G. Arslan, and J. Shamma, "Joint strategy fictitious play with inertia for potential games," *IEEE Trans. Automat. Control*, vol. 54, no. 2, pp. 208-220, Feb. 2009.
[26] J. Zheng, Y. Cai, Y. Liu, Y. Xu, B. Duan, and X. Shen, "Optimal power allocation and user scheduling in multicell networks: Base station cooperation using a game-theoretic approach," *IEEE Trans. Wireless Commun.*, vol. 13, no. 12, pp. 6928-6942, Dec. 2014.
[27] S. Hart and A. Mas-Colell, "A simple adaptive procedure leading to correlated equilibrium," *Econometrica*, vol. 68, no. 5, pp. 1127-1150, 2000.
[28] P. Sastry, V. Phansalkar, and M. Thathachar, "Decentralized learning of Nash equilibria in multi-person stochastic games with incomplete information," *IEEE Trans. Syst., Man, Cybern.*, vol. 24, no. 5, pp. 769-777, May 1994.
[29] J. Zheng, Y. Cai, N. Lu, Y. Xu, and X. Shen, "Stochastic game-theoretic spectrum access in distributed and dynamic environment," *IEEE Trans. Veh. Tech.*, vol. 64, no. 10, pp. 4807-4820, Oct. 2015.
[30] Y. Xu, J. Wang, Q. Wu, A. Anpalagan, and Y. Yao, "Opportunistic spectrum access in unknown dynamic environment: A game-theoretic stochastic learning solution," *IEEE Trans. Wireless Commun.*, vol. 11, no. 4, pp. 1380-1391, Apr. 2012.
[31] M. Bennis, S. M. Perlaza, P. Blasco, Z. Han, and H. V. Poor, "Self-organization in small cell networks: A reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 12, no. 7, pp. 3202-3212, Jul. 2013.
[32] T. Innovations, "LTE in a nutshell," White Paper, 2010.
[33] T. S. Rappaport, *Wireless Communications: Principles and Practice (2nd ed.).* Prentice Hall PTR, 2001.
[34] Bernard Sklar, "Rayleigh fading channels in mobile digital communication systems Part I: Characterization". *IEEE Commun. Mag.*, vol. 35, no. 7, pp. 90-100, Jul. 1997.
[35] E. Cuervo, A. Balasubramanian, D. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, "MAUI: Making smartphones last longer with code offload," in *Proc. ACM Eighth Int. Conf. Mobile Syst., Appl., Serv.*, 2010, pp. 49-62.
[36] B. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "Clonecloud: Elastic execution between mobile device and cloud," in *Proc. ACM Sixth Conf. Comput. Syst.*, 2011, pp. 301-314.
[37] S. Dey, Y. Liu, S. Wang, and Y. Lu, "Addressing response time of cloud-based mobile applications," in *Proc. 1st Int. Workshop Mobile Cloud Comput. Netw.*, 2013, pp. 3-10.
[38] T. Roughgarden, *Selfish Routing and the Price of Anarchy*. Cambridge, MA, USA: MIT Press, 2005.
[39] K. S. Narendra and M. A. L. Thathachar, *Learning Automata: An Introduction.* Englewood Cliffs, NJ: Prentice-Hall, 1989.
[40] M. A. L. Thathachar and P. S. Sastry, *Networks of Learning Automata: Techniques for Online Stochastic Optimization.* Springer Science & Business Media, 2011.
[41] O. E. Ayach and R. W. Heath, "Interference alignment with analog channel state feedback," *IEEE Trans. Wireless Commun.*, vol. 11, no. 2, pp. 626-636, Feb. 2012.



Jianchao Zheng (M'16) received his B.S. degree in Communications Engineering, and Ph.D. degree in Communications and Information Systems from College of Communications Engineering, PLA University of Science and Technology, Nanjing, China, in 2010 and 2016 respectively. From 2015 to 2016, he was a Visiting Scholar with the Broadband Communications Research Group, Department of Electrical and Computer Engineering, University of Waterloo, Canada. He is currently a Research Associate with the National Innovation Institute of Defense Technology, Academy of Military Sciences PLA China. He is also a Post-Doctoral Research Associate with the College of Communications Engineering, PLA Army Engineering University. He has published several papers in international conferences and reputed journals in his research area. His research interests include interference mitigation techniques, green communications and computing networks, game theory, learning theory, and optimization techniques.

Yueming Cai (M'05-SM'12) received the B.S. degree in Physics from Xiamen University, Xiamen, China in 1982, the M.S. degree in Micro-electronics Engineering and the Ph.D. degree in Communications and Information Systems both from Southeast University, Nanjing, China in 1988 and 1996 respectively. His current research interest includes cooperative communications, signal processing in communications, wireless sensor networks, and physical layer security.

Yuan Wu (M'10-SM'16) received the Ph.D degree in Electronic and Computer Engineering from the Hong Kong University of Science and Technology, Hong Kong, in 2010. He is currently an Associate Professor in the College of Information Engineering, Zhejiang University of Technology, Hangzhou, China. During 2010-2011, he was the Postdoctoral Research Associate at the Hong Kong University of Science and Technology. During 2016-2017, he is the visiting scholar at the Broadband Communications Research (BBCR) group, Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research interests focus on radio resource allocations for wireless communications and networks, and smart grid. He is the recipient of the Best Paper Award in IEEE International Conference on Communications (ICC) in 2016.



Xuemin (Sherman) Shen (M'97-SM'02-F'09) received the B.Sc.(1982) degree from Dalian Maritime University (China) and the M.Sc. (1987) and Ph.D. degrees (1990) from Rutgers University, New Jersey (USA), all in electrical engineering. He is a Professor and University Research Chair, Department of Electrical and Computer Engineering, University of Waterloo, Canada. He is also the Associate Chair for Graduate Studies. Dr. Shen's research focuses on resource management in interconnected wireless/wired networks, wireless network security, social networks, smart grid, and vehicular ad hoc and sensor networks. He is an elected member of IEEE ComSoc Board of Governor, and the Chair of Distinguished Lecturers Selection Committee. Dr. Shen served as the Technical Program Committee Chair/Co-Chair for IEEE Globecom'16, Infocom'14, IEEE VTC'10 Fall, and Globecom'07, the Symposia Chair for IEEE ICC'10, the Tutorial Chair for IEEE VTC'11 Spring and IEEE ICC'08, the General Co-Chair for ACM Mobihoc'15, Chinacom'07 and QShine'06, the Chair for IEEE Communications Society Technical Committee on Wireless Communications, and P2P Communications and Networking. He also serves/served as the Editor-in-Chief for IEEE Network, Peer-to-Peer Networking and Application, and IET Communications; an Associate Editor-in-Chief for IEEE Internet of Things Journal, a Founding Area Editor for IEEE Transactions on Wireless Communications; an Associate Editor for IEEE Transactions on Vehicular Technology, Computer Networks, and ACM/Wireless Networks, etc.; and the Guest Editor for IEEE JSAC, IEEE Wireless Communications, IEEE Communications Magazine, and ACM Mobile Networks and Applications, etc. Dr. Shen received the Excellent Graduate Supervision Award in 2006, and the Outstanding Performance Award in 2004, 2007, 2010, and 2014 from the University of Waterloo, the Premiers Research Excellence Award (PREA) in 2003 from the Province of Ontario, Canada, and the Distinguished Performance Award in 2002 and 2007 from the Faculty of Engineering, University of Waterloo. Dr. Shen is a registered Professional Engineer of Ontario, Canada, an IEEE Fellow, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, and a Distinguished Lecturer of IEEE Vehicular Technology Society and Communications Society.