

SDN-based Resource Allocation in Edge and Cloud Computing Systems: An Evolutionary Stackelberg Differential Game Approach

Jun Du, *Member, IEEE*, Chunxiao Jiang, *Senior Member, IEEE*, Abderrahim Benslimane, *Senior Member, IEEE*, Song Guo, *Senior Member, IEEE* and Yong Ren, *Senior Member, IEEE*

Abstract—Recently, the boosting growth of computation-heavy applications raises great challenges for the Fifth Generation (5G) and future wireless networks. As responding, the hybrid edge and cloud computing (ECC) system has been expected as a promising solution to handle the increasing computational applications with low-latency and on-demand services of computation offloading, which requires new computing resource sharing and access control technology paradigms. This work establishes a software-defined networking (SDN) based architecture for edge/cloud computing services in 5G heterogeneous networks (HetNets), which can support efficient and on-demand computing resource management to optimize resource utilization and satisfy the time-varying computational tasks uploaded by user devices. In addition, resulting from the information incompleteness, we design an evolutionary game based service selection for users, which can model the replicator dynamics of service subscription. Based on this dynamic access model, a Stackelberg differential game based cloud computing resource sharing mechanism is proposed to facilitate the resource trading between the cloud computing service provider (CCP) and different edge computing service providers (ECPs). Then we derive the optimal pricing and allocation strategies of cloud computing resource based on the replicator dynamics of users' service selection. These strategies can promise the maximum integral utilities to all computing service providers (CPs), meanwhile the user distribution can reach the evolutionary stable state at this Stackelberg equilibrium. Furthermore, simulation results validate the performance of the designed resource sharing mechanism, and reveal the convergence and equilibrium states of user selection, and computing resource pricing and allocation.

Index Terms—Edge/cloud computing; software-defined networking (SDN), resource pricing and allocation, evolutionary game; Stackelberg differential game.

I. INTRODUCTION

Recently, computation-heavy applications are experiencing a dramatic increasing over the Fifth Generation (5G) and future wireless networks. There is evidence that such applications, including mining process for Proof-of-Work (PoW) in blockchain, interactive gaming, virtual reality, video services, etc., have become premier drivers of the exponential computing task growth [1], [2], [3]. To handle such increasing

computing requirements, hybrid edge and cloud computing (ECC) systems have been expected to provide low-latency and on-demand computing services to users [4], [5], [6]. In ECC systems, cloud computing, as the traditional solution of computation offloading for user devices, is usually implemented at cloud nodes physically located far from users, which results in a long latency service response. Aiming at this problem, edge computing has been proposed as the complement of cloud computing by enabling users to upload computational tasks to the edge of networks [7], [8], which can eliminate the latency and enhance the reliability of services. However, with the growing amount of computational task requirements, computational power limited edge servers might be overwhelmed with severe performance degradation. A feasible solution for this problem is forwarding these tasks at edge nodes to the remote cloud center [9], [10], which can be considered as computation offloading between edge computing service providers (ECPs) and the cloud computing service provider (CCP). Therefore, to achieve the optimal and stable performance of CCP systems, an efficient cloud computing resource sharing mechanism plays an important role resulting from the constrained resource equipped by the CCP and time-varying user requirements among the CCP system. In addition, such mechanism is more challenging when the dynamic service subscription of users is taken into account [11]. This work will establish a hybrid ECC system, in which users can upload their computational tasks to nearby ECPs or the remote CCP dynamically. In addition, by considering the dynamic service subscription of users among the CCP and ECPs, this work will focus on the computing resource sharing and computation offloading mechanism design in the ECC system to realize an efficient utilization of computing resource and satisfy the service requirements of users.

As mentioned previously, the mobility and time varying service selection of users may bring difficulties to efficient resource sharing mechanism designs. In addition, there always exist bidirectional data interactions, including service subscription, task uploading, service response, etc., between end users and computing servers located at either the CCP or ECPs in the ECC system. These frequent interactions may lead to congestions at different computing providers (CPs) [12]. To solve these problems, an appropriate network architecture is necessary to realize an effective management of the hybrid ECC system. In recent years, software-defined networking (SDN) has been considered as an advanced network architecture to achieve flexible resource management and

J. Du and Y. Ren are with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, P. R. China (e-mail: blgdujun@gmail.com, reny@tsinghua.edu.cn).

C. Jiang is with Tsinghua Space Center, Tsinghua University, Beijing 100084, P. R. China (e-mail: jchx@tsinghua.edu.cn).

A. Benslimane is with the Department of Computer Science, University of Avignon, 84911 Avignon, France (e-mail: abderrahim.benslimane@univ-avignon.fr).

S. Guo is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong (e-mail: song.guo@polyu.edu.hk).

system performance control [13], [14], which can mitigate challenges above. Moreover, taking advantage of the available and accurate information of global system status collected by the SDN controller, the system can make optimal decisions to improve resource utilization and service quality [15]. On the other hand, latency problems, fault and Disruption tolerance, and scalability issues brought by the SDN-based fully centralized control architecture can be well solved by the integrated cloud and edge computing mechanism. Therefore, in this work, an SDN-based architecture will be established for computing resource sharing and computation offloading in the ECC system. With a centralized controller, SDN will help CPs to dynamically adjust the resource sharing and computation offloading strategies, which can match time-varying demands of users by observing their dynamic service selection.

Considering that the SDN-based fully centralized control architecture established in Section II will suffer from latency problems, fault and Disruption tolerance, and scalability issues, this section will introduce an ECC system to realize

A. Related Work

For the integrated ECC system, the computation offloading mechanism plays a crucial role in improving resource utilization and service quality. Such computation offloading involves two aspects. Specifically, in the aspect of users, both the CCP and ECPs can offload users' computational tasks with different processing latency and transfer latency. On the other hand, resulting from the limited computational power equipped, ECPs are not qualified for providing services of heavy-computation tasks processing. Then ECPs have to forward some of these tasks to the remote CCP, which has powerful and dedicated computing resource and can provide services on demand. Such process above can be also considered as computation offloading between the CCP and ECPs. According to such two-layered resource sharing among different CPs and users, how to allocate cloud computing resource among ECPs and users selecting the CCP will influence the resource utilization and service quality significantly.

Driven by the supply and demand of computing resource among the CCP, ECPs and users, the resource trading can be formed and facilitated, which needs to satisfy the demands of users selecting different CPs, and meanwhile maximize the utility of each CP. For these purposes, many researches have focused on effect and efficient resource allocation and sharing mechanisms in edge/cloud systems, by introducing different economic models based on auction [16], [17], contract [18], [19], Stackelberg game [20], [21], [22], etc.. Among these studies, auction and contract based trading mechanisms were designed to motivate participants to report their service requirements or capacities truthfully, which can deal with trustworthiness and information asymmetric issues in the system. On the other hand, Stackelberg game provides a suitable framework to model the interactions of trading strategies made in supply and demand sides, including resource pricing, requests and proving for communications [23], storage [24], energy [25], etc., which can facilitate the resource trading efficiently and dynamically. In [20], a multi-leader

multi-follower Stackelberg game was studied to provide cost-effective migrations of data centers in edge-cloud environment. To optimize resource allocation of all cloud and fog computing nodes, a Stackelberg game was formulated in [21], in which fog computing relied on a set of low-power fog nodes that were located close to the end users to offload the services originally targeting at cloud centers. A two-stage Stackelberg game was introduced into the blockchain consensus process in order to incentive the cooperation between the edge/cloud providers and the miners in a PoW-based blockchain network [22]. Similarly, in some current studies, Stackelberg game frameworks were also formulated to model the interaction between the edge/cloud nodes and users [26], [27]. However, all these studies above only considered the computing resource trading between users and CPs, or between the CCP and ECPs, while interactions and influences among the three levels were hardly investigated. In fact, users' service subscription will impact computation offloading between the CCP and ECPs. In addition, computing service qualities received by users selecting different CPs will vary with different resource sharing strategies made by CPs. Then users will change their selection strategies for better services, considering that users are rational. It is difficult to model and analyze these interactions above, since that the strategies made by the three parties will impact and be impacted by each other. To solve such interactive issues, this work will establish an evolutionary game based model to analyze the users' dynamic service selection among CPs. In addition, we will propose a Stackelberg differential game based cloud computing resource sharing mechanism, which will dynamically determine the optimal resource pricing and allocation/request strategies for the CCP and ECPs. In this Stackelberg differential game, the differential equation is introduced based on the replicator dynamics of user selections, which can establish the connection between evolutionary game operated among users and Stackelberg differential game operated among CPs. According to such hierarchical control and optimization, computing resource utilization and user service quality can be both improved.

As mentioned previously, dynamic user selections will bring challenges to the optimization of resource sharing. Such joint optimization for users and heterogeneous CPs can be implemented efficiently by an SDN-based architecture. The SDN-based architecture design for the ECC systems has attracted researchers' great attention, especially in the 5G heterogeneous environment and various Internet of Things (IoT) applications [28], [29], [30]. To realize efficient and secure resource management, data processing and access control, different SDN-based architectures have been investigated. In [31], authors introduced a tunnel-less SDN scheme for scalable realization of virtual tenant networks across the 5G heterogeneous infrastructure, which could support migrations of software instances among geo-distributed computing resources. To meet requirements of various applications and improve the end-to-end system performance efficiently, a novel integrated framework including SDN, computing, and caching was designed in [32]. In [33], the cooperation among edge computing nodes was investigated, and their interactions were realized by establishing an SDN related mechanism. An SDN-

based control scheme was designed in [34] for a multi-edge-cloud environment involves huge data migrations to realize an efficient traffic flow scheduling. In addition, different SDN-based distributed and layered network architectures were also investigated to operate edge/cloud computing systems with blockchain techniques, which can deal with problems brought by limited bandwidth, high latency, large volume of data, and real-time analysis requirements [35], [36]. In summary, leveraging the SDN-based architecture, flexible management of heterogenous resource and optimal control of system performance can be implemented to support computation-heavy applications. Therefore, this work will design an SDN-based architecture to optimize the computing resource utilization by considering the dynamic users' service subscription.

B. Contributions and Organization

Main contributions of this paper are summarized as follows.

- We establish an SDN-based architecture for computing resource sharing in the ECC system. Taking advantage of SDN controllers which separate the distributed infrastructure and resource management, the dynamic optimal pricing and allocation strategies can be obtained.
- We design a Stackelberg differential game based cloud computing resource sharing, which determines the optimal resource pricing and allocation/request strategies dynamically. Then an open-loop Stackelberg equilibrium is derived as the optimal solution. Comparing with traditional static strategies, the proposed mechanism can achieve higher integral utilities in a time horizon and faster convergence speed of decision making.
- We propose a hierarchical dynamic game framework composed of evolutionary game in the user layer and Stackelberg differential game in the edge and cloud layer, which can incentive the cooperation of cloud computing resource sharing. Different from the traditional separated control outside the user layer, this work considers the dynamic service selections of users among edge and cloud resource. Based on this consideration, the user service requirements can be satisfied as well as the edge/cloud computing resource can be utilized efficiently.
- We analyze the performance of the designed computing resource sharing mechanism based on the hierarchical dynamic game. Specifically, the existence and uniqueness of equilibrium of user selections, as well as their evolutionary stable states, are analyzed. In addition, the optimal dynamic pricing and allocation of cloud computing resource are derived based on the replicator dynamics of users' service selection. Furthermore, simulation results validate the performance of the designed resource sharing mechanism, and reveal the convergence and stable states of user selection, resource pricing and resource allocation.

The rest of this paper is organized as follows. The SDN-based architecture for the ECC system is established in Section II. Section III presents the system model and proposed hierarchical game framework. An evolutionary game for service selection of user devices is designed in Section IV, and the Stackelberg differential game based computing resource

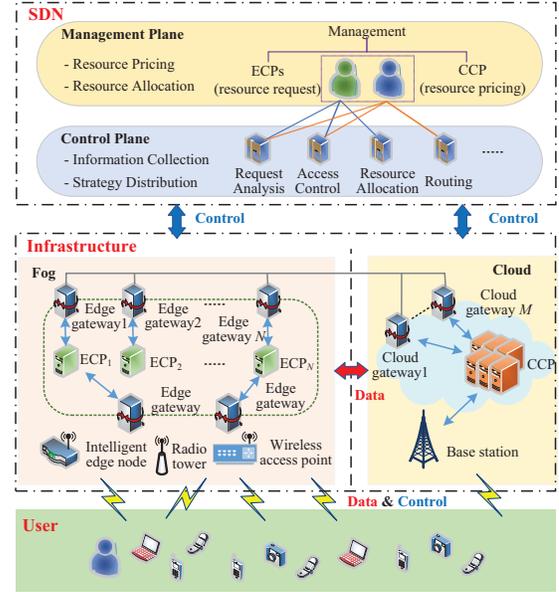


Fig. 1. Architecture of SDN-based resource pricing, sharing and user scheduling in an ECC system.

pricing and allocation schemes are proposed in Section V. Simulations are shown in Section VI, and conclusions are drawn in Section VII.

II. SDN ARCHITECTURE DESIGN FOR EDGE/CLOUD COMPUTING SYSTEMS

This section will propose a model of layered edge/cloud computing service providing system which takes advantage of SDN paradigms, and show that how to implement such cross-layer computing service for users in the designed SDN-based architecture in detail. The SDN-based architecture, which consists of three levels, i.e., the infrastructure plane, control plane and management plane, is established based on the infrastructure in 5G wireless heterogeneous networks (HetNets), as shown in Fig. 1. According to such SDN-based management, CPs will provide edge and cloud computing services to users. In this architecture, the cloud computing center and edge computing nodes are operated by the CCP and different ECPs, respectively. These CPs constitute the infrastructure plane and take charge of providing computing services to different types of user devices. In addition, we consider a user layer outside the SDN architecture, in which user devices such as mobile phones, tablets, etc., can receive computing services by subscribing to different CPs, according to their computational tasks' requirements. Such architecture can realize a dynamic and real-time information collection of task requirements and system workload, and decision making of computation offloading. Next, we will design the function and operation of the three planes in order to satisfy the computing requirements of users, and meanwhile manage the computing resource sharing and providing efficiently.

A. Infrastructure Plane

In the infrastructure plane, the CCP and ECPs provide remote and edge computing services for users, respectively, who can access these CCP and ECPs via edge nodes, such as wireless access points, radio towers, and macro-cell base

station in 5G HetNets. In addition, surveillance cameras and servers at the edge of Radio Access Networks (RANs) can be also operated as edge servers to provide ubiquitous computing services. In current designed mobile computing systems, edge nodes can connect to different ECPs through edge gateways and Device-to-Device (D2D) communication without effecting the backhaul network [12]. By subscribing to these ECPs, users can receive fast response with respect to computing services. Alternatively, users can also select the remote CCP, who usually possesses richer computational power to provide higher-speed computing services than ECPs.

Considering the limited computational power of ECPs, the CCP can share parts of its computational power with ECPs through wireline connections between the edge gateways and cloud gateways. Resulting from the dynamic service selection of users, how to allocate the cloud computing resource among ECPs and users selecting the CCP will influence the efficiency of ECC system. Therefore, an efficient and dynamic control on computing sharing between the CCP and ECPs plays an important role on optimizing the resource utilization and satisfying the resource requirements of users with fast response.

B. Control Plane

As shown in Fig. 1, the SDN-based architecture separates computing resource management from the infrastructure, which forms a hierarchical game based market of cloud computing resource in the control plane. In this plane, information collection of users' subscriptions and strategy distribution for CPs in the infrastructure plane will be implemented through the data information interaction between the infrastructure plane and control plane.

1) *Information Collection*: Through the controller of request analysis, the control plane collects the information of users' subscriptions among different CPs, as well as the local computational power of each CP, and then sends such received information to the management plane in real time. Specifically, the controllers of request analysis and access control are able to communicate with the CCP and ECPs through access points, and then call for the time-varying number of users selecting the corresponding CP.

2) *Strategy Distribution*: The control plane receives pricing and request strategies of cloud computing resource made by the upper management plane, and then distributes these strategies to the CCP and ECPs through the controller of resource allocation. In addition, the access controller and core controller are responsible for resource sharing of cloud resource and access control between edge gateways and cloud gateways, i.e., setting approach network paths between gateways, managing computation offloading and so on.

C. Management Plane

After receiving the information of users' subscription and computational power equipped at each CP, the optimal pricing and request strategies of cloud resource will be determined at the management plane. To be specific, the management will help the CCP to make decisions on dynamic resource pricing, meanwhile help ECPs to determine how much computational power should be requested. Then these decisions will be

returned back to the control plane, and then guide the cloud computing resource sharing between the CCP and ECPs. Such service response above can be implemented fast considering the the wire connections between SDN controllers and edge and cloud gateways, and the powerful processing capacities of SDN servers.

III. SYSTEM MODEL AND HIERARCHICAL GAME FRAMEWORK

Considering that the SDN-based fully centralized control architecture established in Section II will suffer from latency problems, scalability issues, etc., this section will introduce a hierarchical game based computation offloading mechanism to realize real-time and latency-sensitive services close to users.

A. System Model

Consider an ECC system with a finite set $\mathcal{N} = \{1, 2, \dots, N\}$ ECPs overlaying with one remote CCP and providing edge computing service to user devices, including mobile phones, wearable devices, tablets, etc., as shown in Fig. 2. In this work, we consider that the CCP and ECPs can be operated by the mobile network operators (MNOs) or provide the computing service as third parties. These user devices can access and send computational tasks to different ECPs by communicating with edge access points, such as intelligent edge nodes, radio towers, etc., which can upload these computational tasks or service requests to ECPs through edge gateways. In addition, to fulfil some high-complex computational tasks, users can also request the cloud computing resource through base stations and edge access points, which will upload their computational tasks to the CCP. Accordingly, the CCP and ECPs will respond to these computing requests on demand. As compensation, each users needs to pay the CCP or ECPs for accessing fee. Denote p_n as the price charged by ECP n ($n \in \mathcal{N}$) and p_c as the price charged by the CCP, which are fixed access fees per device per unit of time paid by users, considering current charging models set by mobile service operators ¹.

In this system, ECPs, performing as light-weight computing servers, might be deployed at base stations, wireless access points, etc., and then providing computing services with shortened latency. In addition, the CCP, which can promise powerful and stationary computing services, connects to the N ECPs via cloud gateways in a wireline manner. When ECPs cannot fulfill the received computational tasks under their constrained computing resource, they have to forward some part of computational tasks to the remote CCP through wireline backhaul links. Such computation offloading means that ECPs needs to request the CCP for additional computational power. For the shared computing resource and possible energy cost resulting from resource allocation and information exchange for the CCP, ECPs need to pay the CCP with uniform price p in monetized payment unit computational power per unit time. In this work, p is a function of time and can be denoted by $p(t)$. Once announced by the CCP, $p(t)$ can be observed by all ECPs in the system through the SDN architecture.

¹Price p_n and p_c can be set by the MNO or the third-party service providers, depending on who operate these CCP and ECPs.

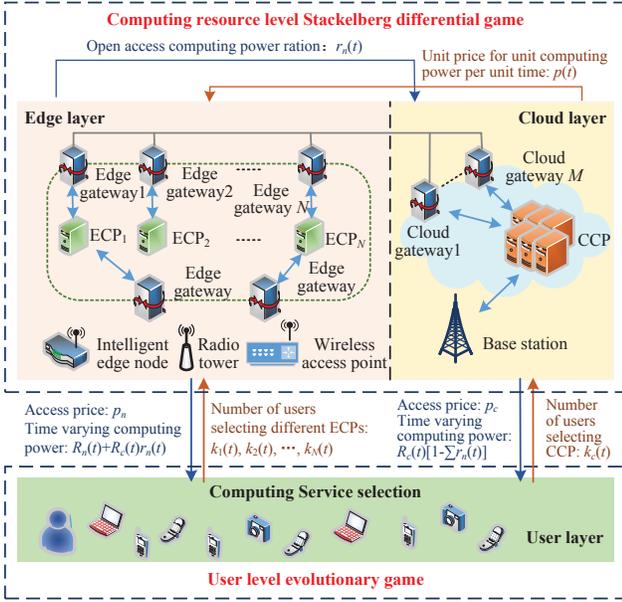


Fig. 2. Hierarchical game based resource pricing and sharing in ECC systems.

After observing price $p(t)$, ECP n ($n \in \mathcal{N}$) decides to request $r_n(t) \in [0, 1)$ proportion of CCPs computational power² during a continuous observation period $[0, T]$. Then the computing resource allocation state of the system can be described by vector $\mathbf{r} = [r_1, r_2, \dots, r_N]^T$. Let $r_c(t)$ denote the current remaining computational power level of the CCP at time t . Then we have $\sum_{n=1}^N r_n(t) + r_c(t) = 1$ for $t \in [0, T]$. In this resource transaction between the CCP and ECPs, the CCP, as the computing resource provider, needs to optimize unit price $p(t)$ to make its resource bring into maximal efficacy; on the other hand, the ECPs, who are the resource receivers and buyers, will make the optimal decision on how many resources to buy to create a tradeoff between its quality of service and cost.

Consider that the local computational power of ECP n is R_n for all $n \in \mathcal{N}$, the total computational power of CCP is R_c ($R_c > R_n, \forall n \in \mathcal{N}$), and the number of all user devices in the system is denoted by K . In addition, let $k_n(t)$ and $k_c(t)$ be the numbers of user devices subscribing to ECP n and the CCP at time t , respectively. Then $x_n(t) = k_n(t)/K$ ($x_n \in [0, 1]$) and $x_c(t) = k_c(t)/K$ represent the population share of ECP n and the CCP accordingly, and we have $\sum_{n=1}^N x_n(t) + x_c(t) = 1$ ³. Based on these definitions above, the computational power allocated to each user selecting ECP n and the CCP can be calculated as

$$\omega_n(\mathbf{x}, \mathbf{r}) = \frac{R_n + R_c r_n(t)}{K x_n(t)}, n \in \mathcal{N}, \quad (1a)$$

$$\omega_c(\mathbf{x}, \mathbf{r}) = \frac{R_c \left(1 - \sum_{n=1}^N r_n(t)\right)}{K x_c(t)}, \quad (1b)$$

²In this work, the computational power is considered as the computing frequency or speed, which can be measured by the computing times per unit time. In blockchain applications, the unit of computational power can be defined as H/s .

³In this work, we only consider the users having the requirement of uploading their computational tasks to the CCP or ECPs. Therefore, there exist at least one computing service provider to be selected by users.

respectively, where vector $\mathbf{x} = [x_1, x_2, \dots, x_N, x_c]^T$ is the population distribution state of the ECC system.

B. Hierarchical Game Framework

Based on the computing resource market model above, the service selection of user devices, pricing strategy of the CCP and computational power requests of ECPs are time-varying and interact with each other. To be specific, if too many users select the same CP, the received computational power for each of these users will decrease according to (1). Then these users tend to leave to other CPs with high average computational power. In addition, CPs (including the CCP and ECPs) with large number of user devices will expect to obtain much computational power to satisfy the computing requirements from accessing user devices. However, such resource request behavior also depends on and will further influence the computing resource price decided by the CCP. To model interactions among users, ECPs and the CCP analyzed above, this work designs a hierarchical dynamic game based scheme to improve the computing service quality and facilitate the computing resource sharing among CPs in the ECC system.

There are two levels in the hierarchical dynamic game designed for the computing service selection, pricing and sharing system, i.e., the user level and the computing resource level. In the user level, the behavior of users' dynamic service selection is formulated and analyzed through an evolutionary game model. Then to model the computing service providing and requirement between the CCP and ECPs, a Stackelberg differential game will be designed to optimize the pricing and sharing strategies for the limited computational power. Such hierarchical dynamic game framework is shown in Fig. 2.

1) *Evolutionary Game in User Level*: According to access price p_c / p_n released by the CCP or different ECPs, as well as the received computational power, each user makes its computing service selection among these CPs to improve its utility. For user devices, the access prices are fixed and stay the same over time. On the other hand, one can notice that the received computational power for users is time varying, which depends not only on the number of user devices accessing the same CP currently, but also the dynamic computing resource sharing of CCP among different CPs. Therefore, without complete information, users can hardly make the optimized service selection globally, i.e., among all CPs and over all time duration. As a response, each user device will learn and adapt its selection strategy gradually. To model this learning and adaptation process, an evolutionary game can be designed to describe and analyze the dynamic user behavior. Through replicator dynamics, all user devices in the ECC system will reach the same individual utility at the equilibrium [37], [38].

2) *Stackelberg Differential Game in Resource Level*: To respond to users' computational task requests on demand, computational power limited ECPs need to buy more computing resource from the CCP. In addition, to compensate the potential loss of accessing users and cost resulting from resource sharing, the CCP will charge corresponding ECP with time-varying unit price $p(t)$ for the provided computational power. Such pricing strategy is dynamic according to the

dynamic computational power sharing/requests and number of accessing user devices at the CCP. Accordingly, given unit price $p(t)$, ECPs control their resource requests dynamically to maximize their own utilities. To analyze such dynamic computing resource pricing of CCP and dynamic resource requests of ECPs, this work establishes a non-cooperate Stackelberg differential game, in the two levels of which, all CPs optimize their own strategies to receive maximized utilities. In this Stackelberg game, the CCP performs as the leader and ECPs are followers. These players in the computing resource level makes their own optimal decisions dynamically according to the time-varying service selection decisions of user devices.

IV. EVOLUTIONARY GAME FOR SERVICE SELECTION OF USER DEVICES

In the ECC system, online users with heavy computational tasks will compete for the limited computing resource by selecting and accessing different CPs. Initially, every user selects a candidate CP randomly or by experience. Then to achieve a better service quality, i.e., large computational power and/or low access price, these users will adapt their selection decisions periodically based on the dynamic received computational power, access price, and the population distribution of all users among different CPs. During this adaptation process, users cannot optimize their selection strategies globally, resulting from the asymmetry of information. Therefore, to improve their utilities, users will learn by imitating those selection strategies with high utilities gradually.

Evolutionary game can be expected as a suitable tool for modeling such learning and imitating process. Thus this section will first formulate the evolutionary game framework and evolutionary strategy adaptation for user selection dynamics. Then the evolutionary equilibrium and evolutionary stable state (ESS) will be investigated for the established model.

A. Evolutionary Game based Service Selection

We first formulate the computing service selection of users among different CPs as an evolutionary game model.

1) *Players*: The set of K user devices in the service area are the players of the evolutionary game.

2) *Strategy*: $\mathcal{S} = \{1, 2, \dots, n, \dots, N, c\}$, where $n \in \mathcal{N}$ indicates selecting ECP n for computing service, and $s = c$ means selecting the CCP directly.

3) *Population States*: The population shares of all ECPs constitute the population distribution state denoted by vector $\mathbf{x} = [x_1, x_2, \dots, x_N, x_c]^T \in \mathbb{X}$, where \mathbb{X} represents the state space which contains all possible population distributions.

4) *Utility*: Given computing resource allocation state $\mathbf{r} = [r_1, r_2, \dots, r_N]^T$ and population distribution state $\mathbf{x} = [x_1, x_2, \dots, x_N, x_c]^T$, the utility function of user device selecting ECP n and the CCP are defined by

$$\pi_n(n, \mathbf{x}, \mathbf{r}) = \frac{\beta \omega_n(\mathbf{x}, \mathbf{r})}{p_n} = \frac{\beta (R_n + R_c r_n(t))}{K p_n x_n(t)}, \quad (2a)$$

$$\pi_c(c, \mathbf{x}, \mathbf{r}) = \frac{\beta \omega_c(\mathbf{x}, \mathbf{r})}{p_c} = \frac{\beta R_c \left(1 - \sum_{n=1}^N r_n(t)\right)}{K p_c x_c(t)}, \quad (2b)$$

respectively, $\beta > 0$ is a constant denoting the mapping factor.

5) *Replicator Dynamic*: The replicator dynamic reflects the evolutionary behavior of the population among different strategies, i.e., selecting different CPs over time. In this work, we introduce the definition in [37], [39], [40], and then give the replicator dynamic as follows,

$$\dot{x}_n(t) = \delta x_n(t) [\pi(n, \mathbf{x}(t), \mathbf{r}(t)) - \pi(\mathbf{x}(t), \mathbf{x}(t), \mathbf{r}(t))], \quad n \in \mathcal{N}; \quad (3a)$$

$$\dot{x}_c(t) = \delta x_c(t) [\pi(c, \mathbf{x}(t), \mathbf{r}(t)) - \pi(\mathbf{x}(t), \mathbf{x}(t), \mathbf{r}(t))], \quad (3b)$$

with initial population distribution state $\mathbf{x}(0) = \mathbf{x}_0 \in \mathbb{X} (\forall n \in \mathcal{S})$, where constant $\delta > 0$ is the learning rate of the population which controls the frequency of strategy adaptation for service selection. Moreover, in (3),

$$\pi(\mathbf{x}(t), \mathbf{x}(t), \mathbf{r}(t)) = \sum_{n=1}^N x_n(t) \pi(n, \mathbf{x}(t), \mathbf{r}(t)) + x_c(t) \pi(c, \mathbf{x}(t), \mathbf{r}(t)) \quad (4)$$

is the expected utility of the population given population distribution state $\mathbf{x}(t)$ and computing resource allocation state $\mathbf{r}(t)$. Based on the definitions above, we have

$$\dot{x}_n(t) = \frac{\delta \beta}{K} \left[\frac{R_n + R_c r_n(t)}{p_n} - x_n(t) \left(\sum_{m=1}^N \frac{R_m + R_c r_m(t)}{p_m} + \frac{R_c \left(1 - \sum_{m=1}^N r_m(t)\right)}{p_c} \right) \right], \quad (5a)$$

$$\dot{x}_c(t) = \frac{\delta \beta}{K} \left[\left(\frac{R_c}{p_c} - \sum_{m=1}^N \frac{R_c r_m(t)}{p_c} \right) - x_c(t) \left(\sum_{m=1}^N \frac{R_m + R_c r_m(t)}{p_m} + \frac{R_c \left(1 - \sum_{m=1}^N r_m(t)\right)}{p_c} \right) \right]. \quad (5b)$$

According to this replicator dynamics defined above, the number of user devices selecting ECP n will increase when $\pi(n, \mathbf{x}(t), \mathbf{r}(t)) > \pi(\mathbf{x}(t), \mathbf{x}(t), \mathbf{r}(t))$, and vice versa.

B. Existence and Uniqueness of Equilibrium

According to the population replicator dynamic formulated in (3) - (5) and the established hierarchical game framework, there exists interactions between decisions of computing service selection made by users and computing resource pricing/allocation controls of CPs. In other words, the evolution of population distribution state defined in (3) is controlled by the pricing strategy of the CCP and resource requests of ECPs. Next, Theorem 1 presents that under these controls, there exists the unique population distribution state $\mathbf{x}(t)$ that constitutes the solution of (5).

Theorem 1. *Consider a dynamic service selection system with a fixed population. For the evolutionary behavior of users among different strategies defined as (3) with initial condition $\mathbf{x}(0) = \mathbf{x}_0$, if resource allocation vector $\mathbf{r}(t)$ is a vector of measurable functions on $[0, \infty)$, then there exists the unique population distribution state $\mathbf{x}(t)$ constitute the solution of (5) for all $t \in [0, \infty)$.*

Proof. Given population distribution state $\mathbf{x}(t)$ and computing resource allocation state $\mathbf{r}(t)$, let $f_n(\mathbf{x}(t), \mathbf{r}(t))$ be the right side of (3), i.e.,

$$f_n(\mathbf{x}(t), \mathbf{r}(t)) \triangleq \delta x_n(t) [\pi(n, \mathbf{x}(t), \mathbf{r}(t)) - \pi(\mathbf{x}(t), \mathbf{x}(t), \mathbf{r}(t))]. \quad (6)$$

Given a fixed t , the partial derivative of $f_n(\mathbf{x}(t), \mathbf{r}(t))$ with respect to $\mathbf{x}(t)$ is continuous. In addition, if $\mathbf{r}(t)$ is measurable on $[0, \infty)$, then $f_n(\mathbf{x}(t), \mathbf{r}(t))$ is also measurable for fixed $x_n(t)$ on the same interval. Furthermore, when given any closed bounded set $\Delta \in \mathbb{X}$ and closed interval $[a, b] \in [0, \infty)$, there always exists a positive I to construct an integrable function [37], [41], [42] by

$$I_n(t) = \left| \frac{\delta\beta(R_n + R_c r_n(t))}{K p_n} \right| + I |\Theta|, \quad (7)$$

where

$$\Theta = \frac{\delta\beta}{K} \left[\sum_{n=1}^N \frac{R_n + R_c r_n(t)}{p_n} + \frac{R_c \left(1 - \sum_{n=1}^N r_n(t)\right)}{p_c} \right]. \quad (8)$$

Obviously, it holds that $|f_n(\mathbf{x}(t), \mathbf{r}(t))| \leq I_n(t)$ and $|\partial f_n(\mathbf{x}(t), \mathbf{r}(t)) / \partial x_n(t)| \leq I_n(t)$, for all $(\mathbf{x}, t) \in \Delta \times [a, b]$. Therefore, we have $|f_n(\mathbf{x}^*(t), \mathbf{r}(t)) - f_n(\mathbf{x}(t), \mathbf{r}(t))| = \Theta |\mathbf{x}^*(t) - \mathbf{x}(t)|$. Denote $\Theta_m = \max\{\Theta\}$, then we can further derive that

$$|f_n(\mathbf{x}^*(t), \mathbf{r}(t)) - f_n(\mathbf{x}(t), \mathbf{r}(t))| \leq \Theta_m |\mathbf{x}^*(t) - \mathbf{x}(t)|, \quad (9)$$

which implies that $f_n(\mathbf{x}(t), \mathbf{r}(t))$ satisfies the global Lipschitz condition. According to the analysis above, we can conclude that the solution to this dynamical population evolutionary system under controls of the CCP and ECPs is unique and exists globally [37], [43], [44], [45]. This completes the proof of Theorem 1. \square

C. Analysis of Evolutionary Stable State (ESS)

Consider a situation where a small proportion of user devices switching to a different mixed strategy $\mathbf{y} \neq \mathbf{x}$. Then these user devices can be regarded as mutants of the population. Denote the size of these mutants by a normalized value $\varepsilon \in (0, 1)$. Then the population state after mutation can be given by $(1 - \varepsilon)\mathbf{x} + \varepsilon\mathbf{y}$ [37]. According to definitions above, we first give the definition of *Evolutionary Stable Strategy (ESS)* in Definition 1.

Definition 1. (Evolutionary Stable Strategy) A strategy \mathbf{x}^* is an ESS, if $\forall \mathbf{x} \neq \mathbf{x}^*$, there exist some $\varepsilon_{\mathbf{x}} \in (0, 1)$ ⁴ such that $\forall \varepsilon \in (0, \varepsilon_{\mathbf{x}})$, the following inequality holds.

$$\pi(\mathbf{x}^*, (1 - \varepsilon)\mathbf{x}^* + \varepsilon\mathbf{x}, \mathbf{r}) > \pi(\mathbf{x}, (1 - \varepsilon)\mathbf{x}^* + \varepsilon\mathbf{x}, \mathbf{r}), \quad (10)$$

where $\pi(\mathbf{x}^*, (1 - \varepsilon)\mathbf{x}^* + \varepsilon\mathbf{x}, \mathbf{r})$ and $\pi(\mathbf{x}, (1 - \varepsilon)\mathbf{x}^* + \varepsilon\mathbf{x}, \mathbf{r})$ are the expected utilities of non-mutants and mutants, respectively.

Considering that the ESS is the best response to the evolutionary system, then an ESS is also a Nash Equilibrium (NE).

⁴ $\varepsilon_{\mathbf{x}}$ represents the maximum proportion of users selecting mutant strategies that can be resisted by the ESS. A large $\varepsilon_{\mathbf{x}}$ indicates that the ESS is robust.

In addition, the evolutionary stability of ESS provides a string refinement of the NE. Moreover, in the NE, a single user cannot benefit through deviating from the equilibrium strategy. On the contrary, the ESS can avoid the deviation behavior of a set of players. Next, we summarize that the evolutionary service selection of users presents the globally asymptotical stability converging to the ESS in Theorem 2.

Theorem 2. Consider a dynamic computing resource selection system with a fixed population. For the evolutionary behavior of the population among different strategies defined as (3) with any initial condition $\mathbf{x}(0) = \mathbf{x}_0$ ($x_n, x_c \in (0, 1)$, $\forall n \in \mathcal{S}$), the replicator dynamics for resource selection is globally asymptotically stable and converges to the ESS of game.

Proof. Considering $\mathbf{x} = [x_1, x_2, \dots, x_N, x_c]^T$ and the replicator dynamic derived in (5), we have $\dot{\mathbf{x}} = \mathbf{\Pi}\mathbf{x} + \boldsymbol{\pi}_o$, where $\mathbf{\Pi}$ is a matrix with dimensional $(N + 1) \times (N + 1)$, and

$$\boldsymbol{\pi}_o = \left[\frac{\delta\beta(R_1 + R_c r_1(t))}{K p_1}, \dots, \frac{\delta\beta(R_N + R_c r_N(t))}{K p_N}, \frac{\delta\beta R_c \left(1 - \sum_{m=1}^N r_m(t)\right)}{K p_c} \right]^T. \quad (11)$$

Therefore, the characteristic function of (5) can be derived as $\det(\gamma\mathbf{I} - \mathbf{\Pi}) = (\gamma + \Theta)^{N+1} = 0$, where \mathbf{I} is the identity matrix and Θ is determined by (8). In addition, (8) implies that $\Theta > 0$ for all resource allocation states $\mathbf{r}(t)$. Consequently, $\mathbf{\Pi}$ always has $N+1$ negative eigenvalues, which means that the replicator dynamics for service selection is globally asymptotically stable and converges to the ESS of evolutionary game. This completes the proof of Theorem 2. \square

V. STACKELBERG DIFFERENTIAL GAME BASED DYNAMIC COMPUTATIONAL POWER PRICING AND ALLOCATION

The CCP and ECPs need to make the optimal decisions on computational power pricing and the amount of computational power requests, respectively, considering the dynamic service selection of user devices $\mathbf{x}(t)$. For the CCP, decreasing price $p(t)$ might incentivize ECPs to request and buy more remote computing resource, which will increase the sharing of CCP resource. However, the user devices accessing the CCP might then leave for other ECPs since that the amount of their received computational power decreases. On the other hand, for ECPs, increasing the amount of computational power requests will improve the utilities obtained by user devices according to (2), which will attract more users' selection according to the replicator dynamics as (3). Then further increasing number of users assessing will reduce the utility obtained by each user device. To analyze this dynamic and interactive decision making problem and then facilitate the computing resource trading between hierarchical CPs, we formulate a Stackelberg differential game, in which the CCP and ECPs perform as the leader and followers, respectively. To search the optimal strategies, an open-loop Stackelberg equilibrium is analyzed as the solution of the game.

A. Formulation of Stackelberg Differential Game

As shown in Fig. 2, the single CCP and N ECPs perform as the players of the Stackelberg game. Specifically, the CCP, as the game leader, first announces its unit computing resource price $p(t)$, according to which ECPs, who are the followers of the game, then make their responding decisions of resource requests $r_n(t)$. In this work, we assume that both the CCP and ECPs are rational so that they can make the best response to the system states and strategies of other players, and follow the strategies made by SDN controllers. In addition, in the established Stackelberg differential game, the CCP and ECPs are willing to optimize their integral utilities over the time horizon $[0, T]$, but not the current utilities, by dynamically controlling their pricing and request strategies, responsibility.

As the followers of Stackelberg game, ECPs can optimize the amount of requesting computational power when observing the unit price released by the CCP. However, the time-varying strategies of all ECPs during the time horizon cannot be observed by the CCP in the present moment. Therefore, this work considers that the CCP is able to learn and predict the expected best response of ECPs and then make its pricing strategy dynamically. Next, we formulate the maximization problems of integral utility for both the CCP and ECPs.

1) *Maximization of Integral Utility for ECPs:* Consider that the utility of each ECP is composed of economic profits and penalty of resource sharing performance. To be specific, by setting accessing price p_n ($\forall n \in \mathcal{N}$), ECP n obtains the revenue from users selecting to it, which is depends on the number of subscribed users, i.e., $Kp_n x_n(t)$. In addition, when requesting the CCP for proportion of could computational power $r_n(t)$, ECP n will be charged $R_c p(t) r_n(t)$ by the CCP. Moreover, the costs resulting from the mismatch between resource supply and demand are also taken account of by ECP n when optimizing its resource request strategy $r_n(t)$, from the performance aspect. This mismatch can be modeled as the distance between the current computational power requirements from all subscribing user devices and the current total computing resource can be provided after receiving the CCP's sharing resource. In the follower layer of Stackelberg game, each ECP is trying to maximize its own profits while minimize the costs resulting from the mismatch between resource supply and demand. Consequently, the instantaneous utility of ECP n can be given by

$$u_n(r_n(t), \mathbf{x}(t), p(t)) = \eta_1 p_n N x_n(t) - \eta_2 R_c p(t) r_n(t) - \eta_3 [K\varphi x_n(t) - (R_n + R_c r_n(t))]^2, \quad (12)$$

where $\varphi > 0$ is defined as a nominal value of accessible computing rate for all user devices, and η_1 , η_2 and η_3 are positive weight factors. In addition, the third term in (12) reflects the matching between the computing resource requirement and available service capacity.

According to the instantaneous utility function (12), the ECP utility depends not only on the received computational power shared by the CCP, but also the population distribution of user devices among different CPs. Therefore, given the pricing strategy of the CCP, the integral utility maximization problem for ECPs can be established as an optimal control problem

subject to the population state of evolutionary game operated in the user-level, which is given by

$$\begin{aligned} \max_{r_n(t)} \quad & U_n^{\text{int}}(r_n(t), \mathbf{x}(t), p(t)) \\ & = \int_0^T e^{-\rho t} [\eta_1 p_n N x_n(t) - \eta_2 R_c p(t) r_n(t) \\ & \quad - \eta_3 [K\varphi x_n(t) - (R_n + R_c r_n(t))]^2] dt; \end{aligned} \quad (13a)$$

$$\text{s.t. } \dot{x}_n(t) = \delta x_n(t) [\pi(n, \mathbf{x}(t), \mathbf{r}(t)) - \pi(\mathbf{x}(t), \mathbf{x}(t), \mathbf{r}(t))], \forall n \in \mathcal{N}, \quad (13b)$$

$$\dot{x}_c(t) = \delta x_c(t) [\pi(n, \mathbf{x}(t), \mathbf{r}(t)) - \pi(\mathbf{x}(t), \mathbf{x}(t), \mathbf{r}(t))], \quad (13c)$$

$$\mathbf{x}(0) = \mathbf{x}_0, \quad (13d)$$

$$r_n(t) \in \mathcal{R}, \forall n \in \mathcal{N}. \quad (13e)$$

In (13), $\rho > 0$ denotes the discount rate influencing the discount value of future utilities.

2) *Maximization of Integral Utility for CCP:* Similarly, the CCP optimizes its pricing strategy to maximize the profits paid by the subscribed users and the ECPs receiving the CCP's computational power, while minimize the costs resulting from the performance discrepancy. Then we have the instantaneous utility of CCP as follows,

$$\begin{aligned} u_c(p(t), r_c(t), \mathbf{r}(t)) = & \xi_1 p_c N x_c(t) + \xi_2 R_c \sum_{n=1}^N p(t) r_n(t) \\ & - \xi_3 \left[K\varphi x_c(t) - R_c \left(1 - \sum_{n=1}^N r_n(t) \right) \right]^2, \end{aligned} \quad (14)$$

where ξ_1 , ξ_2 and ξ_3 are positive weight factors. Therefore, the integral utility maximization problem for the CCP can be also established as an optimal control problem subject to the population state of evolutionary game operated in the user-level, which can be formulated as

$$\begin{aligned} \max_{p(t)} \quad & U_c^{\text{int}}(p(t), \mathbf{x}(t), \mathbf{r}(t)) \\ & = \int_0^T e^{-\rho t} \left[\xi_1 p_c N x_c(t) + \xi_2 R_c \sum_{n=1}^N p(t) r_n(t) \right. \\ & \quad \left. - \eta_3 [K\varphi x_n(t) - (R_n + R_c r_n(t))]^2 \right] dt; \end{aligned} \quad (15a)$$

$$\text{s.t. } \dot{x}_n(t) = \delta x_n(t) [\pi(n, \mathbf{x}(t), \mathbf{r}(t)) - \pi(\mathbf{x}(t), \mathbf{x}(t), \mathbf{r}(t))], \forall n \in \mathcal{N}, \quad (15b)$$

$$\dot{x}_c(t) = \delta x_c(t) [\pi(n, \mathbf{x}(t), \mathbf{r}(t)) - \pi(\mathbf{x}(t), \mathbf{x}(t), \mathbf{r}(t))], \quad (15c)$$

$$\mathbf{x}(0) = \mathbf{x}_0, \quad (15d)$$

$$p(t) \in \mathcal{R}. \quad (15e)$$

B. Open-Loop Stackelberg Equilibrium Solutions

In this part, we will analyze the open-loop solutions to the optimal computing resource pricing and requesting problems established above in (15) and (13) for the CCP and ECPs, respectively. For these optimization problems, if the CCP and ECPs choose to commit their strategies from outset, their information structure can be seen as an open-loop pattern, and their strategies become functions of the initial state \mathbf{x}_0 ,

\mathbf{r}_0 and time t , for both the CCP and ECPs. Considering the Stackelberg differential game operation, it needs to search the optimal solution for each ECP first for the given CCP's pricing strategy, and then the CCP can make the decision on the computing price based on solutions of resource request strategies. Next, we will first analyze the optimal resource request problem for each ECP (follower) in a finite time period $[0, T]$, and then the optimal pricing strategy for the CCP (leader) will be obtained based on the ECPs' strategies.

In a Stackelberg differential game, an open-loop Stackelberg equilibrium is regarded as the optimal solution [46], [47]. So we first introduce the definitions of optimal control strategies for the CCP and ECPs.

Definition 2. (Optimal Control Strategy) For the CCP, pricing strategy $p^*(t)$ is optimal if the following inequality holds for all feasible control paths $p(t) \neq p^*(t)$.

$$U_c^{\text{int}}(p^*(t), \mathbf{x}(t), \mathbf{r}(t)) \geq U_c^{\text{int}}(p(t), \mathbf{x}(t), \mathbf{r}(t)). \quad (16)$$

Similarly, for ECP n ($\forall n \in \mathcal{N}$), the proportion of computational power request $r_n^*(t)$ is optimal if inequality (17) holds for all feasible control paths $r_n(t) \neq r_n^*(t)$.

$$U_n^{\text{int}}(r_n^*(t), \mathbf{x}(t), p(t)) \geq U_n^{\text{int}}(r_n(t), \mathbf{x}(t), p(t)). \quad (17)$$

Based on the definition of the optimal control strategy, we give the definition of open-loop Stackelberg game equilibrium in Definition 3.

Definition 3. (Open-loop Stackelberg Equilibrium) Strategy profile $\Phi^*(t) \triangleq \{p^*(t), \mathbf{r}^*(t)\}$ constitutes an open-loop Stackelberg equilibrium if $p^*(t)$ and $\mathbf{r}^*(t)$ are the optimal control strategies for the CCP and ECPs, respectively, given others' strategies.

1) *Open-loop Stackelberg Equilibrium of ECPs:* In order to get equilibrium solutions of the optimization problem formulated in (13), we need to establish the Hamiltonian system for each ECP. Then the open-loop equilibrium solutions of optimization problem can be characterized as the Pontryagin's Maximum Principle, which is the necessary conditions to find the candidate optimal strategies. First, we summarize the Pontryagin's Maximum Principle in Definition 4.

Definition 4. (Pontryagin's Maximum Principle for ECPs) A set of controls $\{r_n^*(t)\}$ constitutes an open-loop equilibrium to the optimization problem formulated in (13), and $\mathbf{x}_f^*(t)$ is the corresponding population distribution state trajectory, if there exists a set of costate functions $\Lambda_n(t) = [\lambda_{n1}(t) \ \lambda_{n2}(t) \ \cdots \ \lambda_{nm}(t) \ \cdots \ \lambda_{nN}(t) \ \lambda_{nc}(t)]^5$

⁵Considering that $r_c(t) = 1 - \sum_{n=1}^N r_n(t)$, then element $\lambda_{nc}(t)$ in $\Lambda_n(t)$ can be eliminated.

such that the following relations are satisfied.

$$r_n^*(t) = \arg \max_{r_n(t)} \{u_n(r_n(t), \mathbf{x}^*(t), p(t)) + \Lambda_n(t) \dot{\mathbf{x}}^*(t)\}, \quad (18a)$$

$$\dot{\mathbf{x}}_n^*(t) = \delta x_n^*(t) [\pi(n, \mathbf{x}^*(t), \mathbf{r}^*(t)) - \pi(\mathbf{x}^*(t), \mathbf{x}^*(t), \mathbf{r}^*(t))], \quad (18b)$$

$$\dot{\mathbf{x}}_c^*(t) = \delta x_c^*(t) [\pi(c, \mathbf{x}^*(t), \mathbf{r}^*(t)) - \pi(\mathbf{x}^*(t), \mathbf{x}^*(t), \mathbf{r}^*(t))], \quad (18c)$$

$$\mathbf{x}^*(0) = \mathbf{x}_0^*, \quad (18d)$$

$$\dot{\Lambda}_n(t) = \rho \Lambda_n(t) - \frac{\partial [u_n(r_n(t), \mathbf{x}(t), p(t)) + \Lambda_n(t) \dot{\mathbf{x}}(t)]}{\partial \mathbf{x}^*(t)}. \quad (18e)$$

In this system, the equilibrium solutions for ECPs are the solutions of the differential game. Therefore, these solutions are also constitute the Stackelberg equilibrium for ECPs. Then we first introduce the Hamiltonian system for each ECP as follows. Based on the Pontryagin's Maximum Principle, the Hamiltonian system of ECP n can be given by

$$H_n(r_n(t), p(t), \mathbf{x}(t), \Lambda_n(t), t) \triangleq u_n(r_n(t), \mathbf{x}(t), p(t)) + \Lambda_n(t) \dot{\mathbf{x}}(t), \quad (19)$$

where costate function $\Lambda_n(t)$ is a function associate with population state $\mathbf{x}(t)$, and is defined by (18e). In addition, each element of costate function $\Lambda_n(t)$, i.e., $\Lambda_{nm}(t)$, is the costate variable of ECP n associated with state x_m . Based on the Hamiltonian function defined in (19), the corresponding maximized Hamiltonian function is defined as follows:

$$H_n^*(\mathbf{x}(t), \Lambda_n(t), t) \triangleq \max_{r_n(t)} \{H_n(r_n(t), p(t), \mathbf{x}(t), \Lambda_n(t), t) | r_n(t) \in \mathcal{R}\}. \quad (20)$$

Lemma 1. The optimal computational power rate solutions for ECP n ($\forall n \in \mathcal{N}$) is

$$r_n^*(t) = -\frac{\eta_2}{2\eta_3 R_c} p(t) + \frac{K\varphi x_n(t) - R_n}{R_c} + \frac{1}{2\eta_3 R_c} \frac{\delta\beta}{K} \Lambda_n \mathbf{q}_n(\mathbf{x}), \quad (21)$$

which also constitutes an open-loop Stackelberg equilibrium for ECP n . In (21), $\mathbf{q}_n(\mathbf{x})$ is an N -dimension vector which is given by

$$\mathbf{q}_n(\mathbf{x}) = \frac{1}{p_n} \mathbf{i}_n - \left(\frac{1}{p_n} - \frac{1}{p_c} \right) [x_1 \ x_2 \ \cdots \ x_N]^T, \quad (22)$$

where N -dimension vector \mathbf{i}_n is a standard basis, i.e., its n -th element is 1 and other elements are 0.

Proof. According to the Pontryagin's Maximum Principle for ECPs, the optimal control strategy of optimization problem (13) must also maximize the corresponding Hamiltonian function. Therefore, all candidates' optimal strategies have to satisfy the following necessary optimality conditions:

$$\frac{\partial H_n(r_n(t), p(t), \mathbf{x}(t), \Lambda_n(t), t)}{\partial r_n(t)} = 0. \quad (23)$$

Then plug (2) into (23), and the optimal computing resource

request can be deduced as

$$r_n^*(t) = -\frac{\eta_2}{2\eta_3 R_c} p(t) + \frac{K\varphi x_n(t) - R_n}{R_c} + \frac{1}{2\eta_3 R_c} \frac{\delta\beta}{K} \Lambda_n \mathbf{q}_n(\mathbf{x}). \quad (24)$$

Furthermore, according to (18e), we can calculate all elements of $\Lambda_n(t)$, which can be given by

$$\dot{\lambda}_{nm} = \lambda_{nm} (\rho + \Theta(\mathbf{r}(t))), \quad m \neq n, \quad (25a)$$

$$\dot{\lambda}_{nn} = \lambda_{nn} (\rho + \Theta(\mathbf{r}(t))) - \eta_1 p_n K, \quad (25b)$$

where $\Theta(\mathbf{r}(t))$ is defined in (8). This completes the proof of Lemma 1. \square

According to the optimal solutions summarized in Lemma 1, we can observe that the optimal computational power requests and allocation for ECPs is an decreasing function of pricing $p(t)$ determined by the CCP.

2) *Open-loop Stackelberg Equilibrium of CCP*: Similarly, we can obtain the open-loop equilibrium solutions of (15) for the CCP based on the dynamic optimal control. In particular, with the definition of optimal strategy for the CCP as (16) in Definition 2, the open-loop equilibrium solutions for the CCP can be characterized as the Pontryagin's Maximum Principle for CCP, as summarized in following Definition 5.

Definition 5. (Pontryagin's Maximum Principle for CCP) A set of controls $\{p^*(t)\}$ constitutes an open-loop equilibrium to the optimization problem formulated in (15), and $\mathbf{x}^*(t)$ is the corresponding population distribution state trajectory, if there exist costate functions $\mathbf{M}(t) = [\mu_{c1}(t) \ \mu_{c2}(t) \ \cdots \ \mu_{cN}(t)]^T$ and $\Psi(t) = [\Psi_1(t) \ \Psi_2(t) \ \cdots \ \Psi_N(t)]^T$ such that the following relations are satisfied.

$$p^*(t) = \arg \max_{p(t)} \{H_c(p(t), \mathbf{x}(t), \mathbf{r}(t), \Lambda(t), \mathbf{M}(t), \Psi(t))\}, \quad (26a)$$

$$\dot{x}_n^*(t) = \delta x_n^*(t) [\pi(n, \mathbf{x}^*(t), \mathbf{r}^*(t)) - \pi(\mathbf{x}^*(t), \mathbf{x}^*(t), \mathbf{r}^*(t))], \quad (26b)$$

$$\dot{x}_c^*(t) = \delta x_c^*(t) [\pi(c, \mathbf{x}^*(t), \mathbf{r}^*(t)) - \pi(\mathbf{x}^*(t), \mathbf{x}^*(t), \mathbf{r}^*(t))], \quad (26c)$$

$$\mathbf{x}^*(0) = \mathbf{x}_0, \quad (26d)$$

$$\dot{\mathbf{M}}(t) = \rho \mathbf{M}(t) - \frac{\partial H_c(p(t), \mathbf{x}(t), \mathbf{r}(t), \Lambda(t), \mathbf{M}(t), \Psi(t))}{\partial \mathbf{x}^*(t)}, \quad (26e)$$

$$\dot{\Psi}_n(t) = \rho \Psi_n(t) - \frac{\partial H_c(p(t), \mathbf{x}(t), \mathbf{r}(t), \Lambda(t), \mathbf{M}(t), \Psi(t))}{\partial \Lambda_n(t)}, \quad (26f)$$

where the Hamiltonian function of the CCP is given by

$$\begin{aligned} & H_c(p(t), \mathbf{x}(t), \mathbf{r}(t), \Lambda(t), \mathbf{M}(t), \Psi(t)) \\ &= \xi_1 p_c K x_c(t) + \xi_2 R_c p(t) \sum_{n=1}^N r_n(t) \\ & - \xi_3 \left[K\varphi x_c(t) - R_c \left(1 - \sum_{n=1}^N r_n(t) \right) \right]^2 \\ & + \sum_{n=1}^N \mu_{cn}(t) \dot{x}_n(t) + \sum_{n=1}^N \left(\sum_{m=1}^N \theta_{nm}(t) \dot{\lambda}_{nm}(t) \right), \end{aligned} \quad (27)$$

$$\Lambda(t) = [\Lambda_1(t) \ \Lambda_2(t) \ \cdots \ \Lambda_N(t)]^T \text{ determined by}$$

(18e), $\mathbf{M}(t) = [\mu_{c1}(t) \ \mu_{c2}(t) \ \cdots \ \mu_{cN}(t)]^T$ and $\Psi(t) = [\Psi_1(t) \ \Psi_2(t) \ \cdots \ \Psi_N(t)]^T$, where $\Psi_n(t) = [\theta_{n1}(t) \ \theta_{n2}(t) \ \cdots \ \theta_{nm}(t) \ \cdots \ \theta_{nN}(t)]^T$, are costate functions for the CCP.

By solving optimization problem (26a) based on Hamiltonian function (27), we provide the optimal pricing strategy in Lemma 2

Lemma 2. The optimal computational power pricing solutions for the CCP is

$$p^*(t) \triangleq f_p(\mathbf{x}(t), \Lambda(t), \mathbf{M}(t), \Psi(t), t), \quad (28)$$

where

$$\begin{aligned} & f_p(\mathbf{x}(t), \Lambda(t), \mathbf{M}(t), \Psi(t), t) \\ & \triangleq \frac{1}{2NB(\xi_2 + \xi_3 R_c NB)} \left\{ \xi_2 \sum_{n=1}^N A_n \right. \\ & + 2\xi_3 NB \left[K\varphi \left(1 - \sum_{n=1}^N x_n \right) - R_c \left(1 - \sum_{n=1}^N A_n \right) \right] \\ & + \frac{\delta\beta B}{K} \sum_{n=1}^N \mu_{cn} \left[-\frac{1}{p_n} - x_n \left(-\sum_{n=1}^N \frac{1}{p_n} + \frac{N}{p_c} \right) \right] \\ & \left. + \frac{\delta\beta B}{K} \sum_{n=1}^N \sum_{m=1}^N \theta_{nm} \lambda_{nm} \left(-\sum_{n=1}^N \frac{1}{p_n} + \frac{N}{p_c} \right) \right\}. \end{aligned} \quad (29)$$

This optimal pricing $p^*(t)$ also constitutes an open-loop Stackelberg equilibrium for the CCP.

Proof. According to Lemma 1, optimal response $r_n^*(t)$ of ECP n can be expressed as

$$r_n^*(t) = A_n(\mathbf{x}(t)) - Bp(t), \quad (30)$$

where

$$\begin{aligned} & A_n(\mathbf{x}(t), \Lambda_n(t)) \\ &= \frac{K\varphi x_n(t) - R_n}{R_c} + \frac{1}{2\eta_3 R_c} \frac{\delta\beta}{K} \Lambda_n(t) \mathbf{q}_n(\mathbf{x}), \end{aligned} \quad (31a)$$

$$B = \frac{\eta_2}{2\eta_3 R_c}. \quad (31b)$$

As assumed previously, the CCP can learn and predict the optimal response $r_n^*(t)$ of ECP n , $\forall n \in \mathcal{N}$. Therefore, plugging (30) into the Hamiltonian function of the CCP (27), then the Hamiltonian function of the CCP become a concave function with respect to $p(t)$. Thus the optimal pricing strategy $p^*(t)$ is unique for the CCP, which has to satisfy the following necessary optimality conditions

$$\begin{aligned} & \frac{\partial H_c(p(t), \mathbf{x}(t), \mathbf{r}^*(t), \Lambda(t), \mathbf{M}(t), \Psi(t), t)}{\partial p(t)} \\ & \triangleq \frac{\partial H_c(p(t), \mathbf{x}(t), \Lambda(t), \mathbf{M}(t), \Psi(t), t)}{\partial p(t)} = 0. \end{aligned} \quad (32)$$

Taking the first derivative of $H_c(t)$ with respect to $p(t)$ and

then we have

$$\begin{aligned}
& (2N\xi_2B + 2\xi_3R_cN^2B^2)p^*(t) \\
&= \xi_2 \sum_{n=1}^N A_n \\
&+ 2\xi_3NB \left[K\varphi \left(1 - \sum_{n=1}^N x_n \right) - R_c \left(1 - \sum_{n=1}^N A_n \right) \right] \\
&+ \frac{\delta\beta B}{K} \sum_{n=1}^N \mu_{cn} \left[-\frac{1}{p_n} - x_n \left(-\sum_{n=1}^N \frac{1}{p_n} + \frac{N}{p_c} \right) \right] \\
&+ \frac{\delta\beta B}{K} \sum_{n=1}^N \sum_{m=1}^N \theta_{nm} \lambda_{nm} \left(-\sum_{n=1}^N \frac{1}{p_n} + \frac{N}{p_c} \right). \tag{33}
\end{aligned}$$

Therefore, the optimal pricing strategy denoted by (28) and (29) can be obtained.

Furthermore, according to (26e) and (26f), we can calculate all elements of $\mathbf{M}(t)$ and $\Psi(t)$, and then we obtain

$$\dot{\mu}_{cn} = \mu_{cn}(\rho + \Theta(\mathbf{r}(t))) - \xi_1 p_c K, \quad \forall n \in \mathcal{N}, \tag{34a}$$

$$\dot{\theta}_{nm} = \theta_{nm} \Theta(\mathbf{r}(t)), \quad \forall n, m \in \mathcal{N}, \tag{34b}$$

where $\Theta(\mathbf{r}(t))$ is defined through (8). This completes the proof of Lemma 2. \square

3) *Open-loop Stackelberg Equilibrium Solutions:* According to the optimal resource pricing and allocation strategies described in Lemma 1 and Lemma 2, $p^*(t)$ and $r_n^*(n)$ ($\forall n \in \mathcal{N}$) can be denoted by

$$p^*(t) \triangleq f_p(\mathbf{x}(t), \Lambda(t), \mathbf{M}(t), \Psi(t), t), \tag{35a}$$

$$r_n^*(t) \triangleq f_r(\mathbf{x}(t), \Lambda(t), \mathbf{M}(t), \Psi(t), t), \tag{35b}$$

$$r_c^*(t) = 1 - \sum_{n=1}^N r_n^*(t). \tag{35c}$$

Then substituting (35) into (2), (25), (34a) and (34b), and a dynamic control system composed of population distribution state $\mathbf{x}(t)$ and all costate variables $\Lambda_n(t)$, $\mathbf{M}(t)$ and $\Psi(t)$ can be provided as follows.

$$\dot{\mathbf{x}}^*(t) \triangleq f_x(\mathbf{x}(t), \Lambda(t), \mathbf{M}(t), \Psi(t), t), \tag{36a}$$

$$\dot{\Lambda}_n^*(t) \triangleq f_{\Lambda,n}(\mathbf{x}(t), \Lambda(t), \mathbf{M}(t), \Psi(t), t), \tag{36b}$$

$$\dot{\mathbf{M}}^*(t) \triangleq f_M(\mathbf{x}(t), \Lambda(t), \mathbf{M}(t), \Psi(t), t), \tag{36c}$$

$$\dot{\Psi}^*(t) \triangleq f_\Psi(\mathbf{x}(t), \Lambda(t), \mathbf{M}(t), \Psi(t), t). \tag{36d}$$

The dynamic control system formulated above is a typical two-point boundary value problem (TPBVP) [48], [49]. By solving this problem, optimal controls $\mathbf{x}^*(t)$, $\Lambda_n^*(t)$, $\mathbf{M}^*(t)$ and $\Psi^*(t)$ can be obtained. Based on the optimal solutions of TPBVP, the open-loop Stackelberg game equilibrium $\Phi^*(t) \triangleq \{p^*(t), \mathbf{r}^*(t)\}$ can be further derived.

VI. SIMULATION RESULTS

In this part, we will analyze the service selection behavior based on the evolutionary game, and then use MATLAB2019b to evaluate the performance of proposed computing resource pricing and allocation mechanisms based on the Stackelberg differential game. First of all, we introduce the scenario setup

of the simulations. In the following simulations, we assume a typical ECC system, in which there are a single CCP and multiple ECPs who can access the computing resource of the CCP. These CPs provide edge and cloud computing services to $K = 100$ user devices randomly distributed within the coverage of an ECC system.

A. Evolution of Population Distribution

For the numerical analysis, we first consider the situation of two ECPs, i.e., ECP 1 and ECP 2. The local available computational power of the two ECPs are set as $R_1 = 2kH/s$ and $R_2 = 1kH/s$, and the fixed access prices of the two ECPs are given by $p_1 = 0.3$ and $p_2 = 0.2$, respectively [37]. In addition, the initial population distribution state is set as $\mathbf{x}_0 = [x_1(t), x_2(t), x_c(t)] = [0.3, 0.3, 0.4]$, and the initial computing resource request state of ECPs is set as $\mathbf{r}_0 = [r_1(0), r_2(0)] = [0, 0]$, which means that each ECP serves its users with its own computing resource at the beginning of the time horizon. Consider different sharable cloud computational power of the CCP, i.e., $R_c = 5kH/s$ indicating a service quality with high-computational power and $R_c = 2kH/s$ indicating a service quality with low-computational power. Then the CCP fixes its access price p_c by selecting values in $\{0.5 > \max\{p_1, p_2\}, 0.2 = \min\{p_1, p_2\}\}$, which can reflect different cost performance of CCP for users. Moreover, set the learning rate of users as $\delta = 1$.

Then we first investigate the dynamics of population distribution state and the evolution process of service selection from initial state \mathbf{x}_0 , which is subject to the control of resource pricing and allocation strategies. Considering that the dynamic change of population distribution indicates the service selection adaptation of users, we record the population distribution state $\mathbf{x}(t) = [x_1(t), x_2(t), x_c(t)]$ over time, and the results of which are shown in Fig. 3. Results in Fig. 3 validate that the proportion of users selecting every CP converges to an equilibrium state at which there is no user willing to change its service selection strategy.

Then we analyze the influence of cloud computing capacity on user selection. As presented in Fig 3(a), when $R_c = 5kH/s$, the CCP setting a lower access price ($p_c = 0.2$) tends to attract more users to select its cloud resource directly, meanwhile share less computing resource to ECPs, although it possesses a larger computing capacity. On the contrary, when setting a rather high access price ($p_c = 0.5$), the CCP will share all of its computing resource to ECPs and then drive its subscribed users away to ECPs. In this case, the utility of CCP mainly comes from its sharing resource to ECPs. Then we analyze the resource selection evolution trajectory when CCP is limited with computational power, i.e., $R_c = 2kH/s$, which are shown in Fig 3(b). In this case, one can notice that the proportion of users selecting the CCP at equilibrium when $p_c = 0.2$ is larger than that when $p_c = 0.5$, which reflects the fact that the lower price will attract more users. Moreover, results in Fig 3(b) also imply that when the CCP has limited computing resource, the optimal pricing strategy for the CCP is reserving its resource to serve users directly by setting relatively higher unit price $p(t)$. These results can

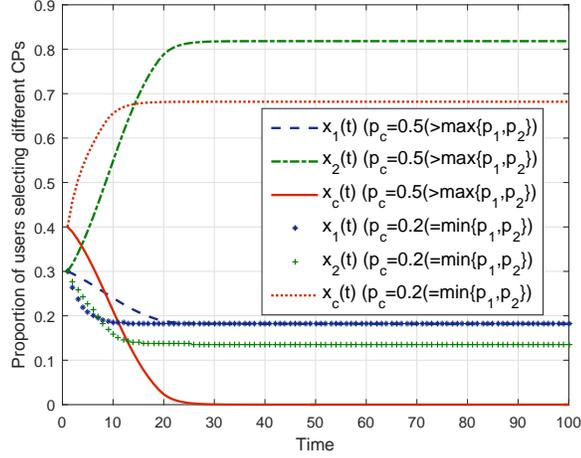
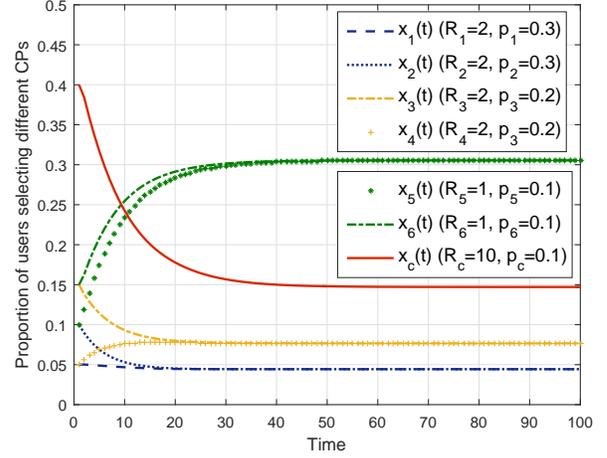
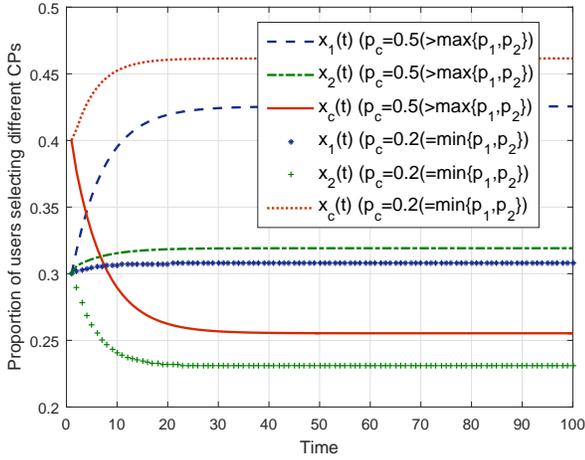
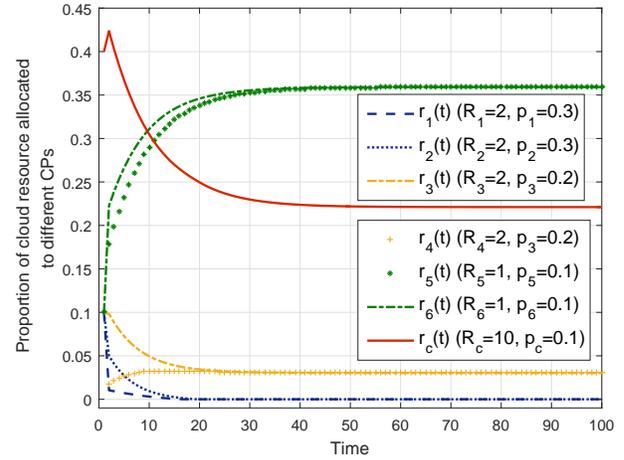
(a) $R_c = 5$ (a) Population distribution x_n, x_c (b) $R_c = 2$ (b) Resource allocation r_n, r_c

Fig. 3. Population distribution state evolution in the ECC system versus different R_c and p_c .

reveal the interaction and influence between the evolutionary game in the user layer and the Stackelberg game in computing resource layer, and validate the rational user behaviors and market rules.

We also test the population distribution dynamic when there are many ECPs in the ECC system. In particular, consider there are $N = 6$ different ECPs selecting the value of their computational power in $\{1, 2\}$ (kH/s) and the value of access price in $\{0.1, 0.2, 0.3\}$. For the CCP, set $R_c = 10kH/s$ and $p_c = 0.1$. In addition, the initial population distribution is set as $\mathbf{x} = [0.05, 0.1, 0.15, 0.05, 0.1, 0.15, 0.4]$. Then we get the evolution of population distribution, cloud resource allocation and user utilities, as shown in Fig. 4. As shown in Fig. 4(a), one can observe that the proportions of users selecting ECPs with the same R_n and p_n simultaneously converge to the same equilibrium from different initial distributions. Moreover, results in Fig. 4(a) also present that the proportion of users selecting ECP 5 and ECP 6 are the highest among all ECPs, which indicates that users are more willing to select the ECPs

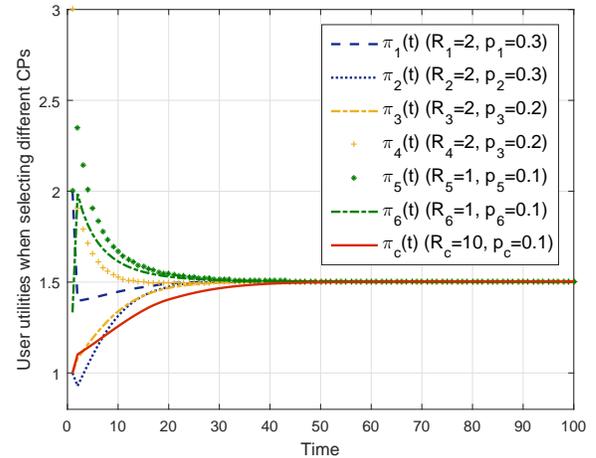
(c) User utilities π_n, π_c

Fig. 4. Evolutions of population distribution, resource allocation and user utilities versus different R_n and p_n when the number of ECPs is $N = 6$.

with lower access price. Next, we investigate how the local computing capacity and access price affect the cloud resource allocation in the ECC system. As shown in Fig. 4(b), ECP 5 and ECP 6 request and receive the most cloud computing

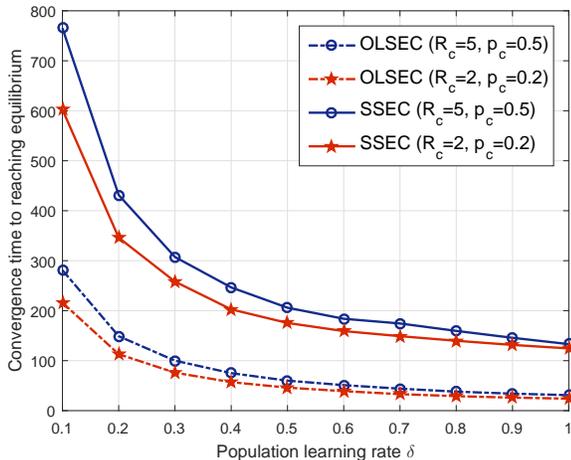


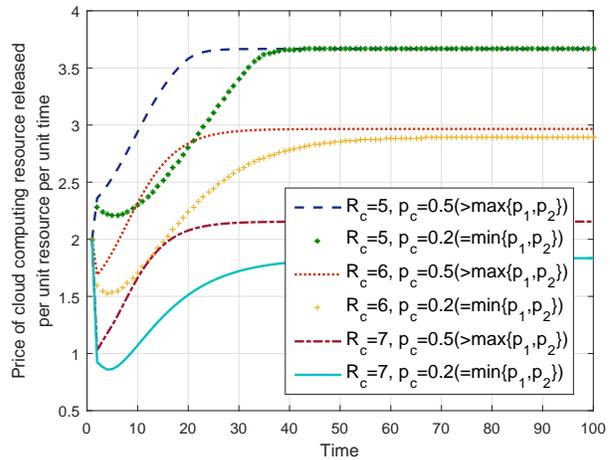
Fig. 5. The convergence time versus increasing learning rate δ and different Stackelberg game equilibrium control schemes.

shares among the six ECPs, and ECP 1 and ECP 2 are allocated the least. Combining the results in Fig. 4(a), results in Fig. 4(b) indicate that ECPs with more population shares tend to request and receive more computing resource from the CCP, which can increase the utilities obtained by the users selecting these ECPs, as formulated in (2), and meanwhile boost the utilities of both the CCP and ECPs. Furthermore, results in Fig. 4(c) validate that through the replicator dynamics, all user devices will reach the same individual utility at the equilibrium.

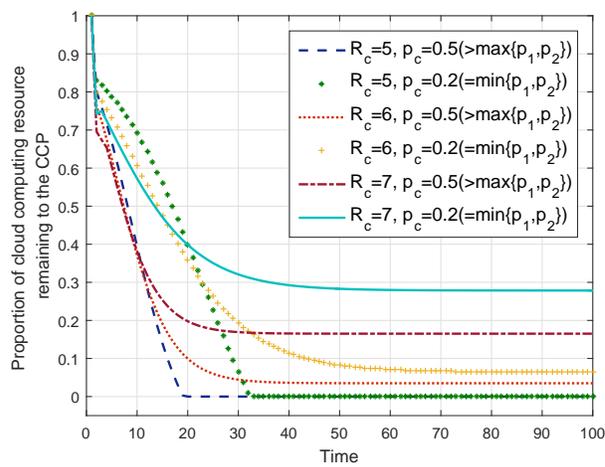
Fig. 5 illustrates the influence of user learning rate δ on the convergence speed of evolutionary game and Stackelberg differential game towards the equilibrium. As defined in (3), learning rate δ controls the frequency of strategy adaption of all users, which will further control the speed of convergence from initial states towards equilibrium. Results shown in Fig. 5 validate that the convergence speed of replicator dynamics grows with the learning rate increasing. In this part of simulation, we also introduce a classic static Stackelberg equilibrium control (SSEC) proposed in [50] and [51] to optimize the resource pricing and allocation strategies. In SSEC, the CCP and ECPs make their decisions only based on the users' selection strategies, but without the considering of dynamic pricing and allocation strategies among CPs. Then results in Fig. 5 indicate that the open-loop Stackelberg equilibrium control (OLSEC) applied in this work can receive a faster convergence speed than SSEC, resulting from the dynamic learning and prediction of all CPs' strategies.

B. Dynamic Pricing and Allocation of Computing Resource

To validate the performance of proposed Stackelberg differential game based resource pricing and allocation strategies, and investigate the impact of R_c and p_c on these strategies made in the computing resource level, we still consider the situation where there are two ECPs in the ECC system. Set $R_1 = 2kH/s$, $R_2 = 1kH/s$, $p_1 = 0.3$ and $p_2 = 0.2$, which are the same as the simulation in Section VI-A. Let



(a) Price $p(t)$



(b) Resource reservation $r_c(t)$

Fig. 6. Dynamic computing resource pricing and allocation in the ECC system versus different R_c and p_c .

R_c select values in $\{5, 6, 7\}$ (kH/s) and p_c choose values in $\{0.5 > \max\{p_1, p_2\}, 0.2 = \min\{p_1, p_2\}\}$.

By applying the evolutionary game based service selection and the Stackelberg differential game based resource pricing and allocation, we obtain the unit price of cloud computing resource and the proportion of cloud computing resource remaining to the CCP, which are shown in Fig. 6(a) and Fig. 6(b), respectively. In Fig. 6(a), results illustrate that the optimal price at equilibrium decreases with increasing total computational power of CCP. Meanwhile, the proportion of cloud computing resource remaining to the CCP at equilibrium increases with growing R_c . In addition, results in Fig 6(a) also indicate that with the same R_c , the optimal price at equilibrium set by the CCP with lower access price p_c is lower than that with high p_c . Combining the results in Fig. 6(b), this trend implies that the utility of CCP with lower access price p_c can be optimized by remaining more cloud computing, which will attract more users selecting the CCP, meanwhile setting a high p_c to reduce the ECPs' willingness of purchasing cloud computing resource. In addition, results in Fig. 6 can also

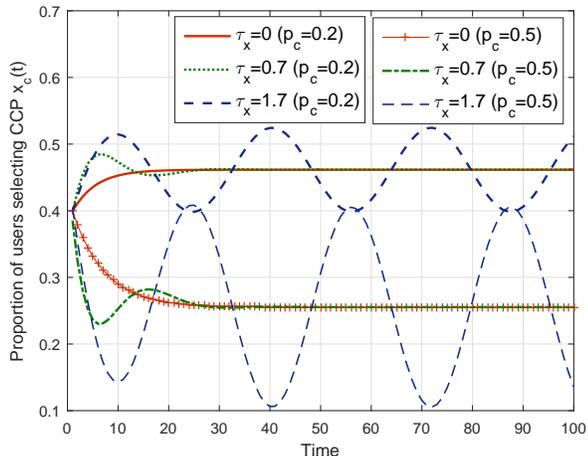


Fig. 7. Proportions of users selecting the CCP under different population delays τ_x and access price of CCP p_c .

validate that the strategies of computing resource pricing and allocation will converge to the Stackelberg equilibrium.

C. Influence of Delay in Replicator Dynamics

Next, we study that how the population distribution states change with delay in replicator dynamics. As defined in (2), the utilities of user devices obtained by selecting different CPs are depends on the service selection strategies of all users in the evolutionary game. However, the information of population distribution state is always delayed resulting from the communication latency. Let $\tau_x \geq 0$ denote the delay of population information. Then the delayed replicator dynamics based on (3) can be given by

$$\dot{x}_{n/c}(t) = \delta x_{n/c}(t - \tau_x) [\pi(n/c, \mathbf{x}(t - \tau_x), \mathbf{r}(t)) - \pi(\mathbf{x}(t), \mathbf{x}(t - \tau_x), \mathbf{r}(t))], \quad (37)$$

where $n \in \mathcal{N}$. Considering that delayed replicator dynamics (37) can be rewritten as $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t - \tau_x) + \mathbf{b}$, then its characteristic equation can be given by $\Theta e^{-\gamma\tau_x} + \gamma = 0$, where Θ has been defined in (8). Here we introduce the necessary and sufficient condition for the stability of delayed replicator dynamics proposed in [52], which can be given by $\tau_x < \pi/2\Theta$. Therefore, the stable ESS can be guaranteed with a small population delay. In this simulation, we set $\tau_x = 0.7$ and $\tau_x = 1.7$ to test different levels of population delay. Other parameters are set as $R_1 = 2kH/s$, $R_2 = 1kH/s$, $R_c = 2$, $p_1 = 0.3$, $p_2 = 0.2$ and $p_c = 0.2$. By applying the delayed replicator dynamics, the proportions of users selecting the CCP are shown in Fig. 7. Results in Fig. 7 validate that the population distribution state can still converge to the equilibrium after dynamic fluctuation, when τ_x is small. On the contrary, when τ_x is large, the equilibrium cannot be reached.

VII. CONCLUSION

In this paper, an SDN-based architecture has been established for edge and cloud computing services in 5G wireless HetNets, which can support efficient and on-demand computing resource management to optimize resource utilization

and complete the time-varying computational tasks uploaded by user devices. In addition, considering the incompleteness of information, an evolutionary game based service selection was designed for users, which can model users' replicator dynamics of service subscription when they request the CCP or ECPs for computing resource. To complete these time-varying computational tasks from users, a Stackelberg differential game based cloud computing resource sharing mechanism was proposed to facilitate the resource trading between the CCP and different ECPs. Moreover, open-loop Stackelberg equilibrium solutions for the CCP (leader) and ECPs (followers), i.e., the optimal resource pricing and allocation strategies, were derived and obtained, which can promise the maximum integral utilities of the leader and followers over the time horizon, respectively. Simulation results have validated the performance of the designed resource sharing mechanism, and revealed the convergence and stable states of user selection, resource pricing, and resource allocation in the ECC system.

REFERENCES

- [1] C. Jiang, H. Zhang, Y. Ren, Z. Han, K.-C. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Commun.*, vol. 24, no. 2, pp. 98–105, Apr. 2017.
- [2] D. T. Nguyen, L. B. Le, and V. K. Bhargava, "A market-based framework for multi-resource allocation in fog computing," *IEEE/ACM Trans. Networking*, vol. 27, no. 3, pp. 1151–1164, Jun. 2019.
- [3] C.-S. Yang, R. Pedarsani, and A. S. Avestimehr, "Communication-aware scheduling of serial tasks for dispersed computing," *IEEE/ACM Trans. Networking*, vol. 27, no. 4, pp. 1330–1343, Aug. 2019.
- [4] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. Networking*, vol. 24, no. 5, pp. 2795–2808, Oct. 2016.
- [5] Y. Jiao, P. Wang, D. Niyato, and K. Suankawmanee, "Auction mechanisms in cloud/fog computing resource allocation for public blockchain networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 9, pp. 1975–1989, Sept. 2019.
- [6] J. Du, C. Jiang, J. Wang, Y. Ren, and M. Debbah, "Machine learning for 6G wireless networks: Carry-forward-enhanced bandwidth, massive access, and ultrareliable/low latency," *IEEE Veh. Technol. Mag.*, vol. 15, no. 4, pp. 123–134, Dec. 2020.
- [7] G. Premsankar, M. Di Francesco, and T. Taleb, "Edge computing for the internet of things: A case study," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1275–1284, Apr. 2018.
- [8] S. Jošilo and G. Dán, "Decentralized algorithm for randomized task allocation in fog computing systems," *IEEE/ACM Trans. Networking*, vol. 27, no. 1, pp. 85–97, Feb. 2019.
- [9] Y. Liu, F. R. Yu, X. Li, H. Ji, and V. C. Leung, "Distributed resource allocation and computation offloading in fog and cloud networks with non-orthogonal multiple access," *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 12 137–12 151, Dec. 2018.
- [10] J. Du, J. Song, Y. Ren, and J. Wang, "Convergence of broadband and broadcast/multicast in maritime information networks," *Tsinghua Sci. Technol.*, vol. 26, no. 5, pp. 592–607, 2021.
- [11] Z. Laaroussi, R. Morabito, and T. Taleb, "Service provisioning in vehicular networks through edge and cloud: an empirical analysis," in *2018 IEEE Conf. Standards Commun. Networking (CSCN)*. Paris France, 29–31 Oct. 2018.
- [12] R. Chaudhary, N. Kumar, and S. Zeadally, "Network service chaining in fog and cloud computing for the 5G environment: data management and security challenges," *IEEE Commun. Mag.*, vol. 55, no. 11, pp. 114–122, Nov. 2017.
- [13] Z. Cao, S. S. Panwar, M. Kodialam, and T. Lakshman, "Enhancing mobile networks with software defined networking and cloud computing," *IEEE/ACM Trans. Networking*, vol. 25, no. 3, pp. 1431–1444, Jun. 2017.
- [14] J. Du, C. Jiang, H. Zhang, Y. Ren, and M. Guizani, "Auction design and analysis for SDN-based traffic offloading in hybrid satellite-terrestrial networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 10, pp. 2202–2217, Oct. 2018.

- [15] C. Liang, Y. He, F. R. Yu, and N. Zhao, "Enhancing video rate adaptation with mobile edge computing and caching in software-defined mobile networks," *IEEE Trans. Wireless Commun.*, no. 10, pp. 7013–7026, Oct. 2018.
- [16] A. Kiani and N. Ansari, "Toward hierarchical mobile edge computing: An auction-based profit maximization approach," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 2082–2091, Dec. 2017.
- [17] Y. Zhang, C.-Y. Wang, and H.-Y. Wei, "Parking reservation auction for parked vehicle assistance in vehicular fog computing," *IEEE Trans. Veh. Technol.*, vol. PP, no. 99, pp. 1–1, Feb. 2019.
- [18] R. Wang, J. Yan, D. Wu, H. Wang, and Q. Yang, "Knowledge-centric edge computing based on virtualized D2D communication systems," *IEEE Communications Magazine*, vol. 56, no. 5, pp. 32–38, May 2018.
- [19] J. Du, E. Gelenbe, C. Jiang, H. Zhang, and Y. Ren, "Contract design for traffic offloading and resource allocation in heterogeneous ultra-dense networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2457–2467, Nov. 2017.
- [20] G. S. Aujla, N. Kumar, A. Y. Zomaya, and R. Rajan, "Optimal decision making for big data processing at edge-cloud environment: An SDN perspective," *IEEE Trans. Ind. Inform.*, vol. 14, no. 2, pp. 778–789, Feb. 2018.
- [21] H. Zhang, X. Yong, S. Bu, D. Niyato, R. Yu, and H. Zhu, "Computing resource allocation in three-tier IoT fog networks: a joint optimization approach combining stackelberg game and matching," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1204–1215, Oct. 2017.
- [22] Z. Xiong, S. Feng, D. Niyato, W. Ping, and H. Zhu, "Edge computing resource management and pricing for mobile blockchain," *IEEE Internet Things J.*, vol. Early Access, 24 Sept. 2018.
- [23] G. S. Aujla, R. Chaudhary, N. Kumar, J. J. P. C. Rodrigues, and A. Vinel, "Data offloading in 5G-enabled software-defined vehicular networks: A stackelberg-game-based approach," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 100–108, Aug. 2017.
- [24] T. Sanguanpuak, D. Niyato, N. Rajatheva, and M. Latva-aho, "Radio resource sharing and edge caching with latency constraint for local 5G operator: Geometric programming meets stackelberg game," *IEEE Trans. Mobile Computing*, pp. 1–1, Early Access, Oct. 2019.
- [25] G. S. Aujla, M. Singh, N. Kumar, and A. Y. Zomaya, "Stackelberg game for energy-aware resource allocation to sustain data centers using RES," *IEEE Transactions on Cloud Computing*, vol. 7, no. 4, pp. 1109–1123, Oct.-Dec. 2019.
- [26] S. Feng, Z. Xiong, D. Niyato, and P. Wang, "Dynamic resource management to defend against advanced persistent threats in fog computing: A game theoretic approach," *IEEE Trans. Cloud Computing*, vol. Early Access, 31 Jan. 2019.
- [27] S. H. Kim, S. Park, C. Min, and C. H. Youn, "An optimal pricing scheme for the energy efficient mobile edge computation offloading with OFDMA," *IEEE Commun. Lett.*, vol. 22, no. 9, pp. 1922–1925, Sept. 2018.
- [28] A. Molina Zarca, J. Bernal Bernabe, I. Farris, Y. Khettab, T. Taleb, and A. Skarmeta, "Enhancing IoT security through network softwarization and virtual security appliances," *Int. J. Network Manage.*, vol. 28, no. 5, p. e2038, Jul. 2018.
- [29] I. Farris, T. Taleb, Y. Khettab, and J. Song, "A survey on emerging SDN and NFV security mechanisms for IoT systems," *IEEE Commun. Surveys & Tutorials*, vol. 21, no. 1, pp. 812–837, Firstquarter 2019.
- [30] G. S. Aujla and N. Kumar, "MEnSuS: An efficient scheme for energy management with sustainability of cloud data centers in edge-cloud environment," *Future Generation Computer Syst.*, vol. 86, pp. 1279–1300, Sept. 2018.
- [31] R. Bruschi, F. Davoli, P. Lago, and J. F. Pajo, "A multi-clustering approach to scale distributed tenant networks for mobile edge computing," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 3, pp. 499–514, Mar. 2019.
- [32] Q. Chen, F. R. Yu, T. Huang, R. Xie, J. Liu, and Y. Liu, "Joint resource allocation for software-defined networking, caching, and computing," *IEEE/ACM Trans. Networking*, vol. 26, no. 1, pp. 274–287, Feb. 2018.
- [33] A. C. Baktir, A. Ozgovde, and C. Ersoy, "How can edge computing benefit from software-defined networking: A survey, use cases & future directions," *IEEE Commun. Surveys & Tutorials*, vol. 19, no. 4, pp. 2359–2391, Jun. 2017.
- [34] G. S. Aujla, N. Kumar, A. Y. Zomaya, and R. Ranjan, "Optimal decision making for big data processing at edge-cloud environment: An sdn perspective," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 778–789, Feb. 2018.
- [35] P. K. Sharma, S. Singh, Y. S. Jeong, and J. H. Park, "Distblocknet: A distributed blockchains-based secure SDN architecture for IoT networks," *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 78–85, Sept. 2017.
- [36] P. K. Sharma, S. Rathore, Y. S. Jeong, and J. H. Park, "SoftEdgeNet: SDN based energy-efficient distributed network architecture for edge computing," *IEEE Commun. Mag.*, vol. 56, no. 12, pp. 104–111, Dec. 2018.
- [37] K. Zhu, E. Hossain, and D. Niyato, "Pricing, spectrum sharing, and service selection in two-tier small cell networks: A hierarchical dynamic game approach," *IEEE Trans. Mobile Computing*, vol. 13, no. 8, pp. 1843–1856, Aug. 2014.
- [38] J. Du, C. Jiang, K.-C. Chen, Y. Ren, and H. V. Poor, "Community-structured evolutionary game for privacy protection in social networks," *IEEE Trans. Inf. Forens. Security*, vol. 13, no. 3, pp. 574–589, Mar. 2018.
- [39] P. D. Taylor and L. B. Jonker, "Evolutionary stable strategies and game dynamics," *Math. Biosci.*, vol. 40, no. 1-2, pp. 145–156, Jul. 1978.
- [40] J. Romero, A. Sanchis-Cano, and L. Guijarro, "Dynamic price competition between a macrocell operator and a small cell operator: A differential game model," *Wireless Commun. Mobile Computing*, vol. 2018, May 2018.
- [41] T. L. Friesz, "Dynamic optimization and differential games," *Alkalmaz. mat. lapok*, vol. 135, no. 1-2, pp. 203–209, 2007.
- [42] Y. C. Ho, "Differential games, dynamic optimization, and generalized control theory," *J. Optimization Theory & Applicati.*, vol. 6, no. 3, pp. 179–209, Sept. 1970.
- [43] C. Jiang, Y. Chen, Y. Gao, and K. R. Liu, "Joint spectrum sensing and access evolutionary game in cognitive radio networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 5, pp. 2470–2483, May 2013.
- [44] F. Li, H. Yao, J. Du, C. Jiang, and Y. Qian, "Stackelberg game-based computation offloading in social and cognitive industrial internet of things," *IEEE Trans. Ind. Inform.*, vol. 16, no. 8, pp. 5444–5455, Aug. 2020.
- [45] C. Jiang, Y. Chen, K. R. Liu, and Y. Ren, "Renewal-theoretical dynamic spectrum access in cognitive radio network with unknown primary behavior," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 3, pp. 406–416, Mar. 2013.
- [46] M. Simaan and J. B. Cruz, "On the stackelberg strategy in nonzero-sum games," *J. Optimization Theory Applicat.*, vol. 11, no. 5, pp. 533–555, May 1973.
- [47] G. Freiling, G. Jank, and S. R. Lee, "Existence and uniqueness of open-loop stackelberg equilibria in linear-quadratic differential games," *J. Optimization Theory Applicat.*, vol. 110, no. 3, pp. 515–544, Sept. 2001.
- [48] G. M. Anderson, "Comparison of optimal control and differential game intercept missile guidance laws," *J. Guidance, Control, Dynamics*, vol. 4, no. 2, pp. 109–115, Mar. – Apr. 1981.
- [49] M. Pachter, E. Garcia, and D. W. Casbeer, "Toward a solution of the active target defense differential game," *Dynamic Games Applicat.*, vol. 9, no. 1, pp. 165–216, Mar. 2019.
- [50] Y. Chen, Z. Jin, and Z. Qian, "Utility-aware refunding framework for hybrid access femtocell network," *IEEE Trans. Wireless Commun.*, vol. 11, no. 5, pp. 1688–1697, Jun. 2012.
- [51] J. B. Clempner and A. S. Poznyak, "Solving transfer pricing involving collaborative and non-cooperative equilibria in nash and stackelberg games: Centralized–decentralized decision making," *Computational Econ.*, no. 1, pp. 1–29, Jul. 2018.
- [52] K. Gopalsamy, *Stability and Oscillations in Delay Differential Equations of Population Dynamics*. Berlin, Germany. Springer, Mar. 1992.