Resources Sharing in 5G Networks: Learning-Enabled Incentives and Coalitional Games

Li Yu, Zongpeng Li, *Senior Member, IEEE*, Jiangchuan Liu, *Fellow, IEEE*, and Ruiting Zhou, *Member, IEEE*

***Abstract*—Smart systems are often battery-constrained, and compete for resources from remote clouds, which results in high delay. Collaboratively sharing resource among neighbors in proximity is promising to control such delay for time-sensitive applications. Rather few existing studies focus on the design between ubiquitous cooperation and competition with learning-enable incentives. In this article, intelligent algorithms are introduced in a distributed fashion, which encapsulates cooperation and competition to coordinate the overall goal of the cellular system with individual goals of Internet of Things (IoT) devices. First, the utility function of the cell and IoT users are designed, respectively. For the former, an incentive mechanism is constructed, where a novel deep actor-critic learning algorithm is developed with a prioritized queue for continuous action space in the differentiated decision-making procedure. For the latter, the energy model is taken into account. Furthermore, the coalition game combined with deep Q-learning framework is explored so as to model and incentivize the cooperation and competition process. Theoretical analysis and simulation studies demonstrate that the improved algorithms perform better than the original version, and they can converge to a Nash-stable optimal or asymptotically optimal solution.**

***Index Terms*—Coalition game, deep learning, decision-making control, resources sharing, smart Internet of Things (IoT).**

I. Introduction

With the proliferation of Internet of Things (IoT) devices of computing and communication capabilities, mobile applications embedded in such devices are also surging in an explosive growth. Consequently, continuous growth in data traffic is witnessed in the network. In addition, the energy-harvested IoT devices with limited battery are severely constrained to address such heavy traffic. Therefore, it is the energy of user equipments (UEs) such as smartphones, vehicles, and drones that is the main bottleneck for serving users [1].

Fortunately, mobile edge computing (MEC) (i.e., fog/edge cloud servers), offering tremendous computing and storing resources, has brought new functionalities of IoT devices, such as machine to machine or device to device communication. MEC has been widely adopted, which is regarded as an efficient paradigm for coping with massive traffic data for mobile UEs, who can save plenty of energy and other resources like computing and storing resources [2]. Since UEs can be highly mobile, they may encounter problems such as service interrupting upon switching from one cell to another. It is desirable for users to enhance their quality of experience (QoE) by exploiting their neighbors' idle resources instead of cloud's. Sharing spare resources reduces the network delay, including round-trip communication delay, improving user QoE. It also significantly reduces the maintenance overhead of cloud for base stations (BSs).

Nevertheless, each IoT device is subject to its own energy capacity limit, which depletes sooner by helping neighbors to process their computing jobs. In such a system, "free-riding" happens when a user utilizes others' resources without offering its own. A well-designed incentives plan is naturally desired, to encourage individual users to participate in the sharing service.

It remains to be an overwhelming drawback of costlier maintenance for cloud servers [3]. From the network operators' perspective, they anticipate to maximize overall network utility, including resource utilization within their overhead. Therefore, it is desirable to design an incentive mechanism to incentivize the individual users to sacrifice their energy for sharing resource under these special circumstances. For example, the cloud cannot accomplish the UE's traffic before its desired completion time. The cloud servers are overloaded. Compared with the distance between its neighbor and the nearest cloud, the UE's remaining energy is not sufficient to transmit to the cloud.

Extensive studies exist on game-theoretical approaches, which fall into two contrasting categories: cooperative and noncooperative game [4]–[6]. An actor-critic approach, deep deterministic policy gradient (DDPG) [7], for continuous control was introduced by Google's DeepMind. DDPG randomly selects samples from experience replay that uses for caching samples to generate next action. However, the samples with the prioritized order are not taken into consideration [7]–[9].

Existing decision-making policies, to the best of our knowledge, are not adequate to accommodate such dynamics of continuous varying network conditions that are stirred by the tremendous traffic growth. Furthermore, little work exists in

Manuscript received September 5, 2019; revised November 13, 2019; accepted December 6, 2019. This work was supported in part by China Scholarship Council, in part by the National Natural Science Foundation of China under Grants 61502504, and 61628209, in part by the Nature Science Foundation of Hubei Province under Grant 2016CFA030, in part by the Technological Innovation Major Projects of Hubei Province under Grant 2017AAA125, and in part by the Science and Technology Program of Wuhan City under Grant 2018010401011288. *(Corresponding author: Zongpeng Li.)*

L. Yu and Z. Li are with the School of Computer Science, Wuhan University, Hubei 430072, China (e-mail: shileyyu@126.com; zongpeng@whu.edu.cn).

J. Liu is with the School of Computing Science, Simon Fraser University, Burnaby(Metro-Vancouver), BC V5A1S6, Canada (e-mail: jcliu@cs.sfu.ca).

R. Zhou is with the School of Cyber Science and Engineering, Wuhan University, Hubei 430072, China (e-mail: ruitingzhou@whu.edu.cn).

Digital Object Identifier 10.1109/JSYST.2019.2958890

integrating the cooperative-competitive scheme with deep models. We aim to strike a delicate balance between 1) the cooperation focusing on the overall interests of system gains and 2) the competition focusing on the self-interest of power consumption loss. The main challenges are as follows.

1) How does it enable each IoT device to make a sophisticated decision between the overall interests and individual benefits independently?
2) How does it model the coalition game formation between cooperation and competition?
3) How can it combine the deep actor-critic reinforcement learning (DARL) approach of the prioritized sampling model with coalition game so as to coordinate the cellular goals and the individual users?

A novel coalition game algorithm integrated with a deep model of prioritized sampling is introduced to address the challenges above. Besides, a delicate tradeoff between the cooperation and the competition is found out by the stable solution of Nash Equilibrium.

Below are the main contributions of this article.

1) We propose intelligent algorithms that encapsulate cooperation and competition to coordinate the global goal of cellular system with the individual one of IoT devices, which is more applicable to real-world settings.
2) We formulate the UEs' energy model and explore the formation of coalition game, facilitating cooperation and competition.
3) A novel deep actor-critic learning algorithm is developed, which investigates the prioritized sampling so as to provide the differentiated services for different users in decision-making procedure on BS's side.
4) A typical distributed deep Q-learning network (DQN) algorithm with individual energy model is explored. For ease of comparison, we provide an elaborate DQN algorithm with coalition game, which is cost-efficient for a plethora of emerging traffic data on the fly.
5) Corroborated by massive simulations, our proposed algorithms can be capable of converging to a Nash-stable optimal solution or an asymptotically optimal one after theoretical analysis and proof.

The remainder of this article is organized as follows. The related work is reviewed in Section II. Preliminaries and problem formulation are depicted in Section III. Subsequently, a distributed coalition game that is combined with the DARL with the prioritized sampling model is presented in Section IV. Then, the theoretic analytics and extensive simulations of our proposed algorithms are elaborated in Section V. Finally, we conclude the article in Section VI.

II. Related Work

Resource sharing was extensively studied in recent literature. A Stackelberg game between BS system and end-users was employed in a cooperative manner, yielding the maximal revenue of each player [5]. Cao and Cai [6] modeled the competitive channel resource of a mobile cloud as a noncooperative game. This model does not consider device's energy consumption. We emphasize user energy loss in the process of its interaction with other devices and clouds. Energy efficiency in cognitive optimization with imperfect hybrid spectrum sensing was investigated in [10], which did not refer to the learning algorithm. A noncooperative energy game with incomplete information was proposed in [11], which leverages a typical reinforcement learning (RL) approach to search for Nash Equilibrium. Furthermore, the strategy of regularized Lagrange multiplier enabled the algorithm to converge to a unique Nash Equilibrium. Conversely, the multiresource scheduling method was proposed in [12], which the cooperative scheduling mechanism was designed by graph theory. However, the RL method was not covered.

RL [13] has emerged as a revolutionary technique to enable a system to "think" and "learn." The agent of RL interacts with its observed environment that produces a reward (penalty) signal to it. Then, the agent takes a corresponding action under certain policy. In this manner, the learner or agent continues to accumulate its experience reserved in the experience replay so that it can search for the optimal value under some state-action pairs in the next time slot, thereby maximizing its long-term rewards [14]. Chen *et al.* [8] employed RL for traffic optimization in a scalable datacenter network, which leveraged their designed components to make local and global decisions, respectively.

Action schema network was studied in a seminal work [15], which used deep learning to optimize the sharing weights. Chinchali *et al.* [16] leveraged the RL method to schedule traffic in cellular networks, which catered a specific scenario of the high volume applications. The RL method was applied to the problem of cache placement that defined the popularity of accessing a file for discrete and discontinuous action space in [9]. Yu and Neely [17] focused on the power control of energy intake, which is aided by the learning method with the discrete action space. While, as opposed to them, we take the continuous action space on the BS side and do not discretize the space into different levels into account.

The conventional RL approach was modeled as the actor-critic network for regulating the training gap between the estimated value of the main network and the targeted values of the target network in [18]. Similarly, the approach of eligibility trace was developed by [19], which overcame the inherent delay reward of RL. Zhu *et al.* [20] exploited the ϵ-greedy selection to obtain the initial Q-value that is composed of the action, state, and rewards in the experience replay. While, in [21], the Boltzmann probability distribution for making decision was investigated, circumventing the extreme case of ϵ-greedy selection. In this regard, the priority model with the prioritized sampling sequence to choose next action instead of the greedy or stochastic selection is formulated.

III. Preliminaries and Problem Formulation

In this section, the basic knowledge about the DARL approach, network scenario, energy model, and problem formulation are described, respectively. For ease of reference, major notations are summarized in Table I.

A. DARL Approach

Generally, RL is composed of an agent interacting with the learning objective (i.e., environment) in discrete decision

TABLE I
SOME NOTATIONS

Notation	**Description**	**Notation**	**Description**
$\mathbb{V}$	Set of neighbors requesting resources	$\mathbb{C}$	Coalitional partition
e^{max}	Maximum capacity of finite-size battery	U^I	Transferable utility
β	Discount factor and its range is $[0, 1)$	$L[t]$	Amount of cloudlet's workload
ϕ and τ	Parameter of controlling update rate	$\mathbb{K}$	Set of $k \in \{1, 2, \cdots, K\}$ UEs
c	Computing capacity of task for user k (cycles per unit time)	e_{loss}	Energy consumption at unit time
$e_{com}[t]$	Power loss of one's sensing, transmitting and storing traffic	e_{thre}	Threshold of energy in devices
$e_{har}[t]$	Amount of energy harvested at time slot t	ϵ	Greedy selection probability
$e[t]$	Energy stored in the battery at the end of slot $t(t \in \mathbb{T})$	$\mathbb{T}$	Set of $t(t \in \{0, 1, \cdots, T-1\})$ epoch
$\mathbb{K}$	Set of game players competing resources in cloudlet	ζ^π and ζ^Q	Hyper-parameters
μ	Balancing factor between TD error and gradient	σ	Adjusting parameter
ξ	Balancing factor between BS's utility and the individual's	T	Iteration episode (seconds)
$D[t, k]$	Deadline of the requested traffic to be processed for user at t	φ	Hyper-parameters
$\pi[.]$	Actor network with parameters θ^π	$w[t, k]$	UE k's weight at t
$\pi^{'}[.]$	Corresponding target network with parameters $\theta^{\pi^{'}}$	v	A UE's neighbor
$Q[.]$	Critic network with parameters θ^Q	$\|\|D\|\|$	Size of replay buffer D
$Q^{'}[.]$	Corresponding target network with parameters $\theta^{Q^{'}}$	Z	Amount of samples

epochs. The agent (i.e., the device or cellular system) observes the state s_t and takes an appropriate action a_t under certain policy $\pi(s)$ at each decision epoch t. Then, it receives an immediate reward R in an iterative manner. The overarching objective of the agent is to find an optimal policy $\pi(s)$ mapping a state to a deterministic action or a probability distribution over the stochastic actions space according to the maximal value of the cumulative discounted reward under the current state-action pair, which is expressed by

$$Q^*[s_t, a_t] = R[s_t, a_t] + \beta \left[\sum_{s'_t} P[s_t, a_t, s'_t] \max_{a'_t} Q^*[s'_t, a'_t] \right] \tag{1}$$

where $Q^*[s_t, a_t]$ represents the Q-value of the state-action pair $[s_t, a_t]$ under the optimal policy. $P[s_t, a_t, s'_t]$ denotes the agents' transition possibility from current state s_t to next state s'_t for the given action a_t. The name of Q learning is acquired by estimating the Q-value $Q[s_t, a_t]$ of the classic RL algorithm. While the DQN employed the parameterized neural network to find out the optimal Q-function, accommodating the dynamics of network [22].

The procedure of DARL adopted a neural network (or even the deep convolutional neural network abbreviated as DNN) to approximate the Q-value function, which involves the actor part and the critic one. The actor is to search the optimal or suboptimal strategy that is parameterized. Then, it generates actions according to the observed environmental state. Whereas the critic one is to estimate and criticize the current policy by receiving rewards, which, therefore, is called as the estimated Q-function. The evaluated Q-function is guided by the temporal difference (TD) error and trains the corresponding critic network. The TD error is used to reconcile the gap between the actor part and the critic one, diminishing the gap mostly. The actor network, then, uses the output of the critic network to update its parameters of the policy [23].

Recent efforts on employing the DARL approach to achieve the crowdsourcing mechanism of BS such as the study in [24] used the method for continuous control. Subsequently, the endeavors in DeepMind corporation introduced the DDPG approach [7] [25]. The core idea was to coordinate a parameterized actor function $\pi[s_t|\theta^\pi]$ about θ^π with a parameterized critic function $Q(s_t, a_t|\theta^Q)$ with respect to θ^Q. The parameterized actor function returns a Q-value. The parameterized critic function criticized the Q-value how good it is [19]. The parameters θ^π of the actor network that has the actor function $\pi[t]$ can be updated by applying the chain rule to the expected cumulative rewards, both of which are respectively given by [26] and [27]

$$\pi[t] = R[s_t] + \beta\pi[s'_t]|\theta^\pi] \tag{2}$$

$$\begin{aligned} \nabla_{\theta^\pi} J &= \mathbb{E}[\nabla_{\theta^\pi} \log \pi_\theta[a|s] Q[s, a|\theta^Q]|_{s=s_t, a=\pi[s_t|\theta^\pi]}] \\ &\approx \mathbb{E}[\nabla_{\theta^\pi} Q[s, a|\theta^Q]|_{s=s_t, a=\pi[s_t|\theta^\pi]}] \\ &\approx \mathbb{E}[\nabla_a Q[s, a|\theta^Q]|_{s=s_t, a=\pi[s_t]} \cdot \nabla_{\theta^\pi} \pi[s|\theta^\pi]|_{s=s_t}] \end{aligned} \tag{3}$$

where J represents the expected cumulative reward. π is the state value with respect to θ^π. $\mathbb{E}[.]$ is the estimated value. Correspondingly, the critic function in the training network is supervised by the loss function [28] that is denoted as the TD error $\delta_{\theta^Q}[t]$. It is calculated by

$$\delta_{\theta^Q}[t] = \mathbb{E}\left[[Q'[s'_t, a'_t|\theta^Q] - Q[s_t, a_t|\theta^Q]]^2 \right] \tag{4}$$

where $Q'[s'_t, a'_t|\theta^Q]$ is the estimated target value, which is expressed by

$$Q'[s'_t, a'_t|\theta^Q] = R[s'_t, a'_t] + \beta Q[s'_t, \pi[s'_t|\theta^\pi]|\theta^Q]. \tag{5}$$

The DDPG approach carried out the uniform sampling from the experience replay. While it did not take the importance of samples into account. Furthermore, a method with the prioritized sampling was introduced in [29], which demonstrated that the prioritized sampling strategy is superior to the state-of-the-art schemes on game-playing tasks. Each transition sample is endowed a priority regarded as the sequential order of accessing the resources in each epoch so as to fulfill the distinguished service. For example, it is anticipated that the cellular system adopts

DDPG with the prioritized sampling sequence to inspire IoT devices to share their idle resources with each other, preventing the "free-riding behavior" effectively. Therefore, a prioritized sampling method compatible with the cellular utility and the individual one is devised, which not only guarantees the fairness during the competition at BS side but also accounts for the energy consumption of UEs during the cooperation in real-world settings. Analogous to the DDPG algorithm, we employ the parameterized actor-critic network, wherein $\pi^G[s^G[t]|\theta^\pi]$ indicates the parameterized actor function of BS with respect to θ^π. $Q^G[s^G[t], a^G[t]|\theta^Q]$ is the parameterized critic function of BS with respect to θ^Q. BS is the core system in the network model described as follows.

B. Network Model

Consider a software defined network (SDN)-based communication cellular networks with a set $\mathbb{K}$ of k mobile UEs, where SDN has the SDN controller and the OpenFlow switches [30]. The SDN-based cellular core network sets up connection by the Internet or center cloud. The BS with the function of gateway governs hybrid communications for UEs within its coverage and controls its cloud servers (here, it is called a cloudlet). More importantly, BS manages to balance the cloudlet's workload of resources. Mobiles users with energy-constrained power are connected by wireless networks such as Wifi or Bluetooth.

Generally, for the selfish and rational UEs, they are not willing to offer their resources to those who needs them at the expense of depleting energy because of the finite battery. Additionally, the time of processing traffic requested by users is later than the deadline of request, implying that UEs' QoE will be sharply degraded when the cloudlet's workload is overloaded. To this end, BSs are supposed to make an appropriate mechanism of inspiring their UEs to share their idle resources with its neighbors. For example, it is a good way of increasing the weight or the priority of the preemptive resources in cloudlet during the competition so as to encourage the UEs to serve its neighbors.

The mode of UEs competing for cloudlet's resources and cooperating with those who need help is portrayed in Fig. 1. The UEs of A, B, C, and D have different energy in power-limited battery. In the case of the adequate energy, UEs will request some resources such as computing or storing the cloudlet's resources in a competitive manner. BS manages the cloudlets to prioritize the accessing queue. The cloudlet becomes overloaded if its queue turns to be red. The UE C with lower energy will ask its neighbor UE D in a close proximity to it for help. Whether the user D is ready to cooperate with UE C or not depends on the comprehensive measurement covering the energy state, total cost, and the individual payoff, which is modeled subsequently.

C. Energy Model

Since we consider IoT devices with renewable energy by conventional power grid or new rechargeable energy (i.e., solar, radio signal, and so on), the finite-size battery can be viewed as backlog in an energy queue. Provided that $l[t, k']$ stands for the amount of cycles performed by CPU in device, which can be estimated the requested amount of resource for the UE $k'(k' \in \mathbb{K}, k' \neq k)$ at time slot t and can be acquired by exchanging messages. Obviously, the executing time of task can be calculated by $\frac{l[t,k']}{c}$. The residual energy $e_{\text{left}}[t, k]$ of UE k at time t can be estimated by

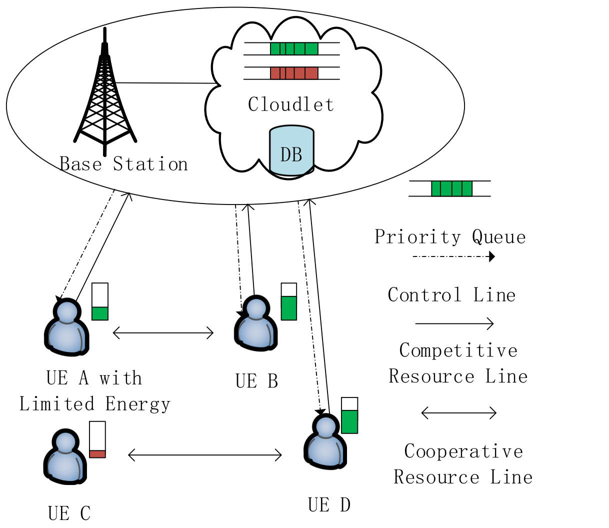

Fig. 1. Ecological mode of cooperation and competition.

$$e_{\text{left}}[t, k] = e[t] + e_{\text{har}}[t] - e_{\text{com}}[t] - e_{\text{ser}}[t, k'] \qquad (6)$$

where $e_{\text{ser}}[t, k'] = (\frac{l[t,k']}{c}) * e_{\text{loss}}$.

After sharing its private resources, the energy left of UE k will follow the constraint of the utilizable energy below

$$e_{\text{left}}[t, k] \geq e_{\text{thre}}. \qquad (7)$$

The energy queue backlog $e[t]$ evolves as follows when the constraint of (7) is satisfied on each slot:

$$e[t+1] = \min\{e_{\text{left}}[t, k], e^{\max}\} \quad \forall t. \qquad (8)$$

Afterwards, the process of formulating problem from the respective of the RL model will be detailed.

D. Problem Formulation

Based on the aforementioned network and energy model, we focus on how to promote the good ecological mode of integrating the cooperation and the competition, taking BS's utility and UEs' gains into consideration. For simplification, assume there are K UEs within the BS's coverage. As for multiple BSs, the design can be further extended for this article. In terms of BS, it manages to determinate how to make a decision of the continuous control and how to boost to share resources mutually so that UE $k(k \in \mathbb{K})$ is willing to cooperate with its neighbor $v(v \in \mathbb{V})$. $\mathbb{V} = \{X_1, X_2, \ldots, X_V\}$. As for the UEs, they are capable of taking proper measures according to the energy left and the revenue of their own.

In general, Markov decision process (MDP) is regarded as the foundation of decision-making models in the learning algorithm, which describes a sequential decision problem [31]. Therefore, the following definition of MDP is given before formulating the model of our problem.

Definition 1 (MDP): An MDP is a tuple encompassing $\{S, A, R, P\}$, where S is the state space; A is the action space of agent; $R : S \times A \to R$ is the reward function mapping state-action pairs to rewards; and $P : S \times A \times S \to [0, 1]$ is the transition function.

Elaborate learning models based on the definition of MDP are formulated from the perspective of BS and UE, respectively.

As for the BS, it operates with unconstrained energy on each slot $t (t \in \mathbb{T})$. Let the tuple of $s^G(L[t], D[t, k]) \in S$ denote its state, where $L[t]$ can be obtained at slot t and $D[t, k]$ is yielded once the traffic packets arrivals in BS. Accordingly, its action space $a^G \in A$ is to assign the weight ratio to each UE k at t, which is calculated by

$$a^G[t, k] = \frac{w[t, k]}{\sum_{k=1}^{K} w[t, k]}. \tag{9}$$

Aiming to express the weight, we measure the importance between TD error and Q gradient, both of which are introduced in Section III-A. The weight is estimated by

$$w[t, k] = \mu \cdot (\delta + \sigma) + (1 - \mu) \cdot \mathbb{E}[|\nabla_{a^G} Q|] \tag{10}$$

where $\mathbb{E}[|\nabla_{a^G} Q|]$ is the average of the absolute values of the Q gradient. σ is a small positive constant for avoiding the edge-cases of transitions that they are not being revisited if σ is zero.

Of note is that the action space is continuous, which is as opposed to [9] and [17] that adopted the discrete action.

Generally speaking, the crowdsourcing platform as the motivator fulfills the incentive mechanism to inspire its users to accomplish their assigned missions [32]. Obviously, it is the employment relationship that is produced between the platform and its users. Nevertheless, unlike [32], the motivator BS and the motivated UEs are a symbiotic relationship that is reflected in rewards between BS and UEs in this article. Specifically, the BS's rewards relies on the UE's accumulated frequency for sharing resources. Its rewards is calculated by

$$R_k^G = \sum_{N_v = X_1}^{X_V} \mathbf{1}_{\{a_k^I = 1\}} * n[N_v, k] \tag{11}$$

where N_v indicates the statistical number of sharing resources of UE k with its neighbor v, which is returned by Algorithm 2 in Section IV-C; let a_k^I denote the action of the individual UE k; $a_k^I = 1$, suggesting the UE k serves its neighbor; and $n[N_v, k]$ is the total times of serving its all neighbors of UE k.

The goal of BS is to search the optimal Q-values $Q_k^{G^*}[s, a]$ of UE k, which is calculated by

$$Q_k^{G^*}[s, a] = R_k^G[s, a] + \beta \left[\sum_{s'} P[s, a, s'] \max_{a'} Q^{G^*}[s', a'] \right]. \tag{12}$$

From the perspective of BS, it anticipates all UEs within its coverage to share their idle resources and to serve its neighbors who need help so as to alleviate the operational overhead of cloudlet and to facilitate the idle resource to be made full use of. That is, the global utility function of BS system is given by

$$U^G = \frac{\sum_{k=1}^{K} Q_k^{G^*}[s, a]}{M[B^G]} \tag{13}$$

where the processing capacity of the cloudlet in BS is a variable denoted as B^G. $M[B^G]$ is the function with respect to B^G, indicating the overhead of maintaining cloudlet and users.

As for UEs based on the energy model defined in Section III-C, let $s^I(e_{\text{left}}[t, k], l[t, v]) \in S$ denote UE k's state space. The action space $a_k^I \in A$ is an indicator function, i.e., $a_k^I \in \{0, 1\}$. The UE k is willing to serve its neighbor v for provisioning the corresponding resource requested if $a_k^I = 1$. Otherwise, the user k does not share its resources. The action of UE k relies on the immediate revenue R_k^I acquired from BS under certain policy at time slot t. Namely, $R_k^I = a^G[t, k]$. Similar to the BS, the long-term goal of individual UE is calculated by

$$Q_k^{I^*}[s, a] = R_k^I[s, a] + \beta \left[\sum_{s'} P[s, a, s'] \max_{a'} Q^{I^*}[s', a'] \right]. \tag{14}$$

Once the network is established, each UE k will build a map of $G\{\mathbb{V}, E\}$, where $\mathbb{V}$ represents the collection of requesting the resources from UE k, which is regarded as the UE k's "ego" network. E stands for the amounts of requesting traffic, which is yielded by exchanging the messages. They expect the allocated weight ratio becomes increasing. Meanwhile, the energy consumption for service is less than that of other neighbors requesting resources under the equivalent amounts of traffic. In other words, the higher the weight ratio is, the more competitive resources in cloudlet they can acquired. It means the completion time of task executed by the cloudlet will be shortened. The UEs can achieve high QoE. Hence, the individual utility function is defined as

$$\begin{aligned} U^I &= \frac{F[w]}{D[e] + 1} * Y_{k,v}[t] \\ \text{s.t.} \quad & D[e] \geq 0 \\ & F[w] \in [0, 1] \\ & Y_{k,v}[t] \in \{0, 1\} \end{aligned} \tag{15}$$

where $F(w) = a^G$. $D(e) = \min e_{\text{ser}}[t, v]$. $Y_{k,v}[t]$ is an indicator function, denoting UE k selects its neighbor v if $Y_{k,v}[t] = 1$. Otherwise, $Y_{k,v}[t] = 0$. Each UE k is supposed to select no more than one neighbor at time slot t in order to have no influence on the resource usage itself.

Herein, the objectives of BS and UEs are formulated, respectively. However, there exist the intractable challenges to be addressed. For instance, it is nontrivial to model the coalition game between the cooperation and the competition. In addition, it is challenging to combine the DARL approach of the prioritized sampling model with the coalition game in order to coordinate the BS's objective and the individual goal.

To this end, an elaborate solution of challenges above in the following section is provided.

IV. Solutions of Challenges

To elucidate how to bridge the gap between the cooperation and the competition and address the challenges above, we elaborate on the coalitional game formation, the deep actor-critic learning approach with the prioritized sampling and the distributed algorithms establishment, respectively.

A. Coalitional Game Formation

Since UE k can acquire more payoffs of its individual utility than that of not forming coalition if it shares resources with its neighbors, UE $k(k \in \mathbb{K})$ is willing to cooperate with its neighbor $v(v \in \mathbb{V}, \mathbb{V} = \{X_1, X_2, \ldots, X_V\}, \mathbb{V} \subseteq \mathbb{K})$. It is evident that there exists $(V+1)$ coalitions formed as $\mathbb{C} = \{C_{x_1}, C_{x_2}, \ldots, C_{x_V}, C_{x_{V+1}}\}$, where $C_{x_z} \bigcap C_{x_{z'}} = \varnothing$, for any $z \neq z'$, and $\bigcup_{z=1}^{V+1} C_{x_z} = K$.

However, because its power consumption is huge if UE k is in cooperation with all neighbors, UE k will not form the grand coalition when taking individual rationality into account. Consider that UE k in the coalitional game can join or leave the coalition after measuring its utility in the physical environment. As a consequence, the coalitional game with the transferable utility is taken into account, which is given as follows.

Definition 2 (Coalition game): Coalition game is viewed as a triple tuple $(\mathbb{K}, U^I, \mathbb{C})$. The coalitional revenues for the individual will be assigned to the $(V+1)$ coalitions after forming the cooperation, i.e., for $\forall C_x \subset \mathbb{C}$ and $\forall z \neq z', \mathbb{C} = \{C_{x_1}, C_{x_2}, \ldots, C_{x_V}, C_{x_{V+1}}\}, C_{x_z} \bigcap C_{x_{z'}} = \varnothing$ and $\bigcup_{z=1}^{V+1} C_{x_z} = K$.

Like [33], assuming that, for any UE $k(k \in \mathbb{K})$, let "$\succeq_k$" denote the preference order, which is defined as a complete, reflexive and transitive binary relation over the set of all coalitions that the user k can possibly form. Hence, each user is able to sort its potential coalitions according to the well-defined preference relation when choosing to join one of the coalitions. The definition of preference order is introduced to estimate the potential coalitions.

Definition 3 (Preference order): For any given UE $k(k \in \mathbb{K})$, $C_x \succeq_k C_{x'}$ represents UE k prefers to join the coalition $C_x(C_x \subset \mathbb{C})$ rather than $C_{x'}(C_{x'} \subset \mathbb{C})$ according to the following estimation of utilitarian order [34]. Namely, for $\forall k, k \in \mathbb{K}$ and $k \in C_x, C_{x'}$, we have $C_x \succeq_k C_{x'} \doteq (U^I[C_x] + U^I[C_{x'} \backslash k]) > (U^I[C_x \backslash k] + U^I[C_{x'}])$, where we use "$\doteq$ " to denote "is defined to be equal to" in this article.

It is observed that the preference relation excludes the case that the UE k is equally willing to be a member of the two coalitions (i.e., C_x and $C_{x'}$) in accordance with Definition 3. The utility sequence implies that the UE k will join in coalition C_x instead of $C_{x'}$ on the premise that the coalition C_x makes the user's utility high when compared with coalition $C_{x'}$. Afterwards, the following transfer trigger to form transferable coalitions is defined.

Definition 4 (Transfer trigger): Based on Definition 2, we fix a partition $\mathbb{C} = \{C_{x_1}, C_{x_2}, \ldots, C_{x_V}, C_{x_{V+1}}\}$ of UE set $\mathbb{K}$, if UE $k(k \in \mathbb{K})$ executes a transfer trigger from C_x to $C_{x'}(C_x \neq C_{x'})$, then the current partition $\mathbb{C}$ is modified into a new partition $\mathbb{C}'$ such that $\mathbb{C}' = \{\mathbb{C} \backslash \{C_x, C_{x'}\}\} \bigcup \{C_x \backslash \{k\}, C_{x'} \bigcup \{k\}\}$.

In nature, it is noted that the transfer trigger is triggered by the preference relation defined in Definition 3. The cooperation among neighbors is set up by the coalition game defined above. Subsequently, the BS employs the DARL approach with the prioritized sampling to carry out the distinctive service for UEs in competitive scenarios.

TABLE II
PARAMETER SETTING IN EXPERIMENTS

Parameters	**Values**
T	10^{-2}
μ	0.6
β	0.99
ξ	0.4
ϵ	0.05
ζ^{π}	0.001
σ, ζ^{Q}, ϕ and φ	0.01

B. DARL Approach With the Prioritized Sampling

The DARL approach employed ϵ-greedy selection [20] or the Boltzmann probability distribution [21] to yield next action in the experience replay. Recent research on DDPG [7] adopted the uniform sampling. Fortunately, the prioritized sampling was explored in [27], in which Xu *et al.* merely considered the importance between the TD-error and the Q gradient for the targeted sampling in the replay. However, it did not take the individual utility of UE into consideration. Besides, the approach in [27] did not apply to our proposed models or scene. Therefore, based on the UE's practical situation such as the energy consumption and payoffs and so on, the priority model weighed by the individual utility of UEs and global utility of BS is developed, enhancing the bridge of the cooperation and competition. The priority is estimated by

$$p = \xi \cdot U^I + (1 - \xi) \cdot U^G \tag{16}$$

where ξ, belonging to (0, 1), indicates the balance factor of fine-tuning the utility between the individual and BS.

C. Distributed Algorithms Establishment

In this section, five algorithms for BS and UEs are elaborated on. A novel DARL algorithm with well-defined priority for sampling is adopted for the continuous control of BS. While the improved DQN algorithm with energy model is developed for the discrete control in the decision-making process in terms of UEs. Moreover, the coalitional game algorithm is designed to boost the cooperative-competitive relationship between BS and UEs.

The DARL framework is outlined in Algorithm 1 first for comparison in simulations. In it, the DARL algorithm employs the double actor-critic networks to further regulate network parameters in the training procedure, i.e., the actor network, the critic network, and the corresponding target networks. Notice that there exist quite a few hyper-parameters in the learning framework. According to the result of empirical study and the validation in extensive simulations, we have found the good settings for some hyper-parameters. The configuration of parameters is outlined in Table II in Section V-E. Of note is that the conventional gradient of the actor-critic network is used for updating network (lines 9–10) with no consideration of the priority. As a result, the DARL algorithm with the prioritized

Algorithm 1: DARL Framework in BS.

```
Input: Initialize $\pi[.]$ with $\theta^{\pi}$ and $Q[.]$ with $\theta^{Q}$;
    Initialize $\pi'[.]$ with $\theta^{\pi'} \doteq \theta^{\pi}$ and $Q'[.]$ with $\theta^{Q'} \doteq \theta^{Q}$;
    Initialize $s^G$, $a^G$ and $\beta$.
Output: $A^G$.
Receive the initial observed state $s_k^G[t]$
for $t = 0$ to $T - 1$ do
    $a_k^G[t] \leftarrow select(A^G)$ by $\epsilon$-greedy algorithm [20]
    Execute $a_k^G[t]$ and observe the immediate reward $R_k^G[t]$
    Put the transition sample
        $(s_k^G[t], a_k^G[t], R_k^G[t], s_k^G[t+1])$ into buffer $D$
    Calculate the targeted value $\pi'[t]$ and $Q'[t]$ :
        $\pi'[s'_t|\theta^{\pi}] = R[s'_t] + \beta\pi[s'_t|\theta^{\pi}]$
        $Q'[s'_t, a'_t|\theta^{Q}] = R[s'_t, a'_t] + \beta Q[s'_t, \pi[s'_t|\theta^{\pi}]|\theta^{Q}]$
    Calculate the gradient of actor network and critic one by (3) respectively: $\nabla_{\theta^{\pi}}\pi$ and $\nabla_{\theta^{Q}}Q^G$
    Calculate TD-error $\delta_{\theta^{\pi}}$ and $\delta_{\theta^{Q}}$ :
        $\delta_{\theta^{\pi}}[t] = \pi'[s'_t|\theta^{\pi}] - \pi[s_t|\theta^{\pi}]$
        $\delta_{\theta^{Q}}[t] = Q'[s'_t, a'_t|\theta^{Q}] - Q[s_t, a_t|\theta^{Q}]$
    Update the parameters of actor network and critic one, respectively:
        $\theta^{\pi}[t+1] \leftarrow \theta^{\pi}[t] + \zeta^{\pi} \cdot \delta_{\theta^{\pi}}[t] \cdot \nabla_{\theta^{\pi}}\pi$; reset
        $\nabla_{\theta^{\pi}}\pi \leftarrow 0$ $\theta^{Q}[t+1] \leftarrow \theta^{Q}[t] + \zeta^{Q} \cdot \delta_{\theta^{Q}}[t]$
        $\cdot \nabla_{\theta^{Q}}Q$; reset $\nabla_{\theta^{Q}}Q \leftarrow 0$
    Update the parameters of the corresponding target network:
        $\theta^{\pi'}[t+1] \leftarrow \phi\theta^{\pi}[t] + (1-\phi)\pi'[t]$
        $\theta^{Q'}[t+1] \leftarrow \phi\theta^{Q}[t] + (1-\phi)Q'[t]$
end for
```

Algorithm 2: DARL Algorithm With the Prioritized Sampling.

```
Input: Initialize $\pi[.]$ with $\theta^{\pi}$ and $Q[.]$ with $\theta^{Q}$;
    Initialize $\pi'[.]$ with $\theta^{\pi'} \doteq \theta^{\pi}$ and $Q'[.]$ with $\theta^{Q'} \doteq \theta^{Q}$;
    Initialize $s^G$, $a^G$ and $\beta$;
    Initialize priority in $D$: $p \leftarrow 0$.
Output: $A^G$.
1:  Receive the initial observed state $s_k^G[t]$
2:  for $t = 0$ to $T - 1$ do
3:      $a_k^G[t] \leftarrow select(A^G)$ by $\epsilon$-greedy algorithm [20]
4:      Execute $a_k^G[t]$ and observe $R_k^G[t]$
5:      Put sample $(s_k^G[t], a_k^G[t], R_k^G[t], s_k^G[t+1])$ into $D$
6:      while $||D|| \neq \varnothing$ do
7:          Calculate $Q[.]$ and $U^G$ by the corresponding (12) and (13) respectively
8:          if (BS has received $U^I$) then
9:              Calculate $\delta_{\theta^{\pi}}[t]$, $\delta_{\theta^{Q}}[t]$, $\nabla_{\theta^{\pi}}\pi$ and $\nabla_{\theta^{Q}}Q$
10:             Calculate the weight ratio of actor and critic network respectively:
                    $w_{\pi} = \mu \cdot (\delta_{\theta^{\pi}}[t] + \sigma) + (1-\mu) \cdot \mathbb{E}[|\nabla_{\theta^{\pi}}\pi|]$
                    $w_{Q} = \mu \cdot (\delta_{\theta^{Q}}[t] + \sigma) + (1-\mu) \cdot \mathbb{E}[|\nabla_{\theta^{Q}}Q|]$
11:             Calculate $p$ according to (16)
12:             Update variation of weight ratio respectively:
                    $\triangle_{w_{\pi}} \leftarrow \triangle_{w_{\pi}} + w_{\pi} \cdot p$
                    $\triangle_{w_{Q}} \leftarrow \triangle_{w_{Q}} + w_{Q} \cdot p$
13:             Update the parameters of actor and critic network respectively:
                    $\theta^{\pi}[t+1] \leftarrow \theta^{\pi}[t] + \zeta^{\pi} \cdot \triangle_{w_{\pi}}$; reset $\triangle_{w_{\pi}} \leftarrow 0$
                    $\theta^{Q}[t+1] \leftarrow \theta^{Q}[t] + \zeta^{Q} \cdot \triangle_{w_{Q}}$; reset $\triangle_{w_{Q}} \leftarrow 0$
14:         else
15:             Select the maximal $U^G$ as next sample
16:         end if
17:     end while
18:     Update the parameters of the corresponding target network respectively: $\theta^{\pi'}$ and $\theta^{Q'}$
19: end for
```

sampling is additionally devised. As illustrated in Algorithm 2, the prioritized sampling is presented (lines 6–17). Note that the measurement between the TD error and the gradient of functions acts as the coefficient of priority (lines 10–12). Furthermore, the user-oriented interface is designed to embody the UE's influence on BS system (line 8). BS can choose the maximal utility according to (13) that will be the candidate of next sample (line 15).

Here the DQN algorithm with energy model and with coalition game on UE's side are illustrated, respectively. For simplicity, the DQN framework is first extracted in Algorithm 3 [22].

The DQN algorithm with energy model for UE is described in Algorithm 4. Observe that each UE needs to choose the appropriate neighbor according to the estimated utility U^I if its residual energy satisfies the energy constraints (lines 4–8). Then, the DQN algorithm is invoked to learn with trial and error (line 14).

In addition, the DQN algorithm with the coalition game for UE is developed in Algorithm 5, in which the cooperative procedure with its neighbors is interpreted. To be specific, the UE (agent) k first judges whether the preference order is satisfied according to Definition 3 (lines 7–14). Second, the coalitional partition is updated iteratively in accordance with Definition 4 until the partition converges to the final Nash-stable (lines 3–15). Finally, combined with the coalitional partition, the individual explores the coalition and judges whether the coalition converges to the optimal or near-optimal Q-value (line 16).

V. Theoretical Analysis and Numerical Result

A. Convergence

In this section, the convergence of the proposed coalition formation algorithm is guaranteed [36].

Theorem 1: Any initial coalition $\mathbb{C}_{\text{ini}}$ in user set $\mathbb{K}$ in Algorithm 5 will always converge to a final coalitional partition $\mathbb{C}_{\text{fin}}$, which includes a number of the disjoint coalitions after it has been undergone a series of transfer trigger operations.

Proof: We observe that each transfer trigger in Algorithm 5 will either yield a new partition according to the result of calculating the preference utility or stay the unchangeable partition. Hence, part of coalitions may degenerate into the sets of few users, and even be empty. There exist at most $(V+1)$ partitions in the network, namely V nodes of requesting resource from UE

Algorithm 3: DQN Framework.

```
Input: Initialize replay buffer $D$;
    Initialize statistical number $N_v$: $N_v \leftarrow 0$;
    Initialize the main DQN with parameters $\varphi$ and
    corresponding target DQN with parameters $\varphi'$, $\varphi' \doteq \varphi$.
Output: $N_v$ and $A^I$.
    /* Return $A^I$ by DQN */
1:  Observe $s^I[t,k]$
2:  if $(Y_{k,v}[t]=1)$ then
3:    Perform $a^I[t,k]=1$
4:    Obtain the immediate reward $R_k^I$ and next
      observation $s^I[t+1]$
5:    Store the experience $(s^I[t,k], a^I[t,k], R_k^I, s^I[t+1])$
      into buffer $D$
6:    Capture $Z$ samples from $D$ by mini-batch gradient
      descent method [35]
7:    Calculate the target Q-value $Q'[t+1]$ from the
      target DQN:
      $Q'[t]=R_k^I[t]+\beta Q[s^I[t+1], \arg\max_{a^{I'}} Q[s^I[t+1], a^{I'}|\varphi]|\varphi']$
8:    Update the parameters of main DQN by minimizing
      the loss function $L[\varphi]$:
      $L[\varphi]=\mathbb{E}[Q'[t]-Q[s[t],a[t]|\varphi]]^2$ and execute a
      gradient descent step on $L[\varphi]$ with respect to $\varphi$
9:    Update the parameters of target DQN with update
      rate $\tau$:
      $\varphi \leftarrow \tau\varphi + (1-\tau)\varphi'$
10: else
11:   $a^I[t,k]=0$
12: end if
13: Update individual energy by (8)
14: $N_v++$
```

Algorithm 4: DQN Algorithm With Energy Model for UE.

```
Input: Initialize $D$ and $N_v$: $N_v \leftarrow 0$;
    Initialize the main DQN with parameters $\varphi$ and
    corresponding target DQN with parameters $\varphi'$, $\varphi' \doteq \varphi$.
Output: $N_v$, $U^I$ and $A^I$.
1:  for $t \leftarrow 1$ to $T$ do
2:    Calculate $e_{\text{left}}[t,k]$ according to (6)
      /* Select the neighbor with maximal utility for UE $k$
      sharing resources */
3:    for $v \leftarrow 1$ to $V$ do
4:      if ((7) holds) then
5:        $Y_{k,v}[t] \leftarrow 1$
6:        Receive the immediate reward from BS
7:        Calculate the utility $U^I[k,v]$ by (15)
8:        Send $U^I[k,v]$ to BS
9:      else
10:       $Y_{k,v}[t] \leftarrow 0$
11:       Send no message to BS
12:     end if
13:   end for
14:   The same statements from line 1 to 14 in Algorithm 3
15: end for
```

Algorithm 5: DQN Algorithm With Coalitional Game for UE.

```
Input: Given any partition $\mathbb{C}_{\text{ini}}$ of neighbor set $\mathbb{V}$ for UE $k$;
    Set current partition as $\mathbb{C}_{\text{cur}} \leftarrow \mathbb{C}_{\text{ini}}$;
    Initialize $D$ and $N_v$: $N_v \leftarrow 0$;
    Initialize main DQN with parameters $\varphi$ and
    corresponding target DQN with parameters $\varphi'$, $\varphi' \doteq \varphi$.
Output: $N_v$, $U^I$ and $A^I$.
1:  for $t \leftarrow 1$ to $T$ do
2:    Calculate $e_{\text{left}}[t,k]$ according to (6)
      /* Select the neighbor with maximal utility for UE $k$
      sharing resources */
3:    repeat
4:      Choose one neighbor $v(v \in \mathbb{V})$ and denote its
        coalition as $C_x(C_x \subset \mathbb{C}_{\text{cur}})$; set $Y_{k,v}[t]=1$
5:      Randomly search for another possible coalition
        $C_x, C_{x'} \subset \mathbb{C}_{\text{cur}}$, where $C_{x'} \neq C_x$; set $Y_{k,v'}[t]=1$
6:      Calculate $U^I[C_x]$ and $U^I[C_{x'}]$
7:      if $(C_{x'} \succeq_k C_x)$ then
8:        UE $k$ leaves its current coalition $C_x$ and joins the
          new coalition $C_{x'}$
9:        $N_v \leftarrow N_v + 1$
10:       Update the current partition set:
          $\mathbb{C}_{\text{cur}} \leftarrow \{\mathbb{C}_{\text{cur}} \backslash \{C_x, C_{x'}\}\} \bigcup \{C_x \backslash \{k\},$
          $C_{x'} \bigcup \{k\}\}$
11:       Update individual energy by (8)
12:     else
13:       $N_v \leftarrow 0$ and set $Y_{k,v'}[t]=0$
14:     end if
15:   until The partition converges to the final Nash-stable
      partition $\mathbb{C}_{\text{fin}}$
16:   The same statements from line 1 to 14 in Algorithm 3
17: end for
```

k in the UE's "ego" network and UE k itself with no coalition. Since the number of partitions for the given user set $\mathbb{K}$ is the Bell number [34]. It is concluded that the sequence of transfer trigger will terminate and converge to a final partition $\mathbb{C}_{\text{fin}}$. ■

B. Stability

It is a crucial factor for system to stay stable. Thereby, the system's stability is evaluated by the following definition.

Definition 5 (Nash-stable structure): A coalitional partition $\mathbb{C}=\{C_{x_1}, C_{x_2}, \ldots, C_{x_V}, C_{x_{V+1}}\}$ is Nash-stable, if $\forall k \in \mathbb{K}$, $k \in C_x$, $C_x \succeq_k (C_{x'} \bigcup \{k\})$ for all $C_x, C_{x'} \subset \mathbb{C}$, $C_{x'} \neq C_x$.

Theorem 2: The final partition $\mathbb{C}_{\text{fin}}$ yielded by Algorithm 5 is Nash-stable and the coalitional game has the Nash-stable coalitional structure if no user can make its contribution to the BS's utility rise by varying its shared resource strategy, namely $C_x^* = \arg\max_{C_x} U^I[C_x]$, $\forall C_x \subset \mathbb{C}$, $\mathbb{C}^* = \{C_{x_1}^*, C_{x_2}^*, \ldots, C_{x_V}^*, C_{x_{V+1}}^*\}$ is the final Nash-stable coalitional structure.

Proof: Contradiction is adopted. Provided that the final formed coalition partition $\mathbb{C}_{\text{fin}}$ is not Nash-stable. There exists a UE $k(k \in \mathbb{K})$ located in one coalition $C_x(C_x \subset \mathbb{C})$ currently.

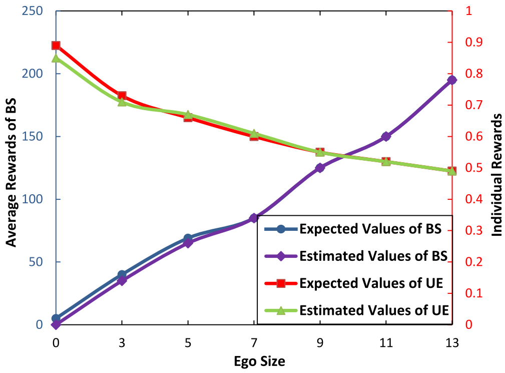

Fig. 2. Comparison of gap between targeted values and estimated values under the average rewards of BS and individual rewards with different ego size.

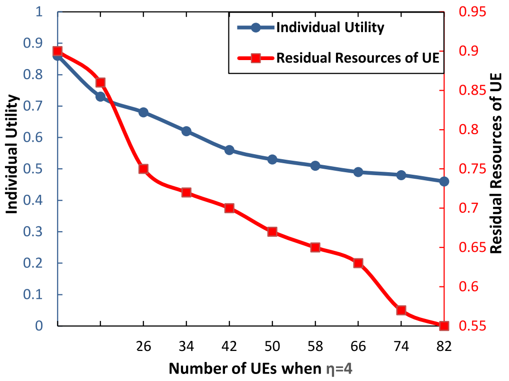

Fig. 3. Impact of individual utility and residual resources on the increasing number of UEs when $\eta = 4$.

If it chooses another coalition $C_{x'}(C_{x'} \subset \mathbb{C})$ according to the preference order $(C_{x'} \bigcup\{k\}) \succeq_k C_x$, then the user leaves the current coalition C_x and joins the new coalition $C_{x'}$, suggesting that the final partition $\mathbb{C}_{\text{fin}}$ should be updated. It contradicts with the definition of $\mathbb{C}_{\text{fin}}$. ■

C. Optimality

Theorem 3: Algorithm 2 and Algorithm 5 will converge to an optimal or asymptotically optimal solution after a limited number of iterations.

Proof: Since the coalition transfer is triggered according to the individual utility, the coalition partition is repeated until the partition is converged to a Nash-stable solution. In addition, the solution is validated by DQN algorithm after the loop is terminated. During the loop, the objective is to find out the optimal solution or approximated optimum. It is observed that the gap between the estimated UE's Q-value and the targeted one is quite small as demonstrated by simulations in Fig. 2 in Section V-E. The improved DARL approach with the prioritized sampling is illustrated in Algorithm 2. The TD error is adopted to guide the training network by the finite iterations. Likewise, as sketched in Fig. 2, the gap of BS's Q-value is still negligible. Hence, it is concluded that the optimal performance of Algorithm 2 and Algorithm 5 can be guaranteed. ■

D. Complexity

Theorem 4: Given T, D, and V, the asymptotically computational complexity of Algorithm 2 and Algorithm 5 is $O(T \times D)$ and $O(T \times V)$, respectively.

Proof: Algorithm 2 is running on the TensorFlow [37], which is appointed graphic processing unit to capture the features of BS system. Thereby, the parallel processing parameters in the double actor-critic networks is applicable. According to Algorithm 2, there exist two-fold loops nested. The computational complexity is approximated as $O(T \times D)$. As for Algorithm 5, the number of coalitions depends on the size of UE k's "ego" network. Assuming that there are V neighbors who need help. V iterations are carried out according to Algorithm 5 (lines 3–14). That is, the computational complexity is estimated by $O(T \times V)$ in the worst case. ■

Generally speaking, the size of buffer D and "ego" network is quite small in our physical applications, i.e., $D, V \ll T$. Hence, it is linear for the asymptotically computational complexity of Algorithm 2 and Algorithm 5.

E. Numerical Result

In this section, massive simulations for corroborating the performance of the proposed algorithms are carried out. The system performance is evaluated under the various parameters in terms of the total system utility (i.e., BS system) and individual utility (i.e., IoT devices). Besides, the deep analysis for the acquired simulation results has been made. Note that the benchmark DDPG algorithm is adopted, i.e., Algorithm 1 for comparison. Algorithm 1 is the core idea of DDPG. Since, to the best of our knowledge, our proposed the learning-enabled incentives with coalitional game for resources sharing in cellular networks is the first work in relative fields, we, therefore, just employ the enhanced algorithms like Algorithm 2, Algorithm 4, and Algorithm 5 to compare the state-of-the-art algorithm DDPG.

In our implementation, a 2-layer fully connected feed-forward neural network is used to serve as the actor and critic network. There were 64 and 32 neurons in the first and second layers, respectively. The Leaky Rectifier [38] acted an as activation function before the final output layer, which utilizes the softmax for activation. The activation function of softmax is to ensure the sum of output values equals one. The empirical settings by extensive simulations are outlined in Table II. The simulation parameters are mostly derived from [39].

Let the size of UE's "ego" network be fixed to 3 and 7 by varying the number of UEs within one cellular system to be 10–82. The number of coalitions η is derived as 4 and 8, respectively. Fig. 3 shows the impact of individual utility and residual resources on the increasing number of UEs when the number of coalitions η is four. Note that the curves of both UE's utility and its residual resources are decreasing as the number of UEs rises when $\eta = 4$. The reason is that the size of queue for computing the cloud's task is limited and that the UEs competing for the resources from cloud are increasing. Thus, the weight is divided by increasing other users. Meanwhile, more UEs are inspired to share their spare resources. Undoubtedly, their utilizable

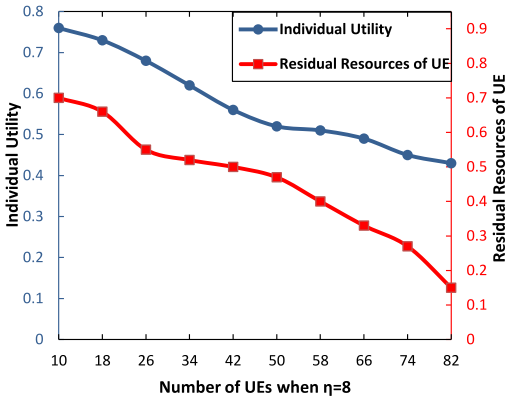

Fig. 4. Impact of individual utility and residual resources on the increasing number of UEs when $\eta = 8$.

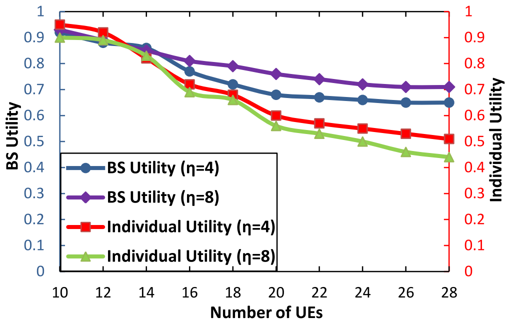

Fig. 5. Relationship between BS utility and individual utility with the increasing UEs when $\eta = 4$ and $\eta = 8$.

resources are fewer and fewer. The individual utility is reduced as the increased number of UEs. Herein, we recall (15) defined in Section III-D. The growth of UEs implies the competition of cloud's resources becomes increasing. Nevertheless, the total sum of allocating weight ratio of each UE is unchanged, meaning $F[w]$ is decreasing. Likewise, the ratio of the residual resources in the individual devices is also diminished since user shares its idle resources to yield more rewards of weight ratio. Once it shares its own resources, it inevitably depletes its restricted energy regarded as one of resources.

Correspondingly, when $\eta = 8$, the impact of the individual utility and the residual resources on the increasing number of UEs is sketched in Fig. 4. It shows the changing curves of the individual utility and resources left under the ever-increasing network scale of UE is constructed after network establishment. It is observed that UE's utility almost keeps stable even though its resources available are reduced by 22.4%, suggesting that its idle resources are sufficiently leveraged by sharing resources with others. According to Algorithm 2, we have acquired that the more the number of times for resource sharing is, the more the rewards are assigned to the user. Therefore, although the curve is slightly declining, the individual utility is hardly changed.

Fig. 5 illustrates the relationship between BS utility and individual utility with the increasing UEs when $\eta = 4$ and $\eta = 8$, respectively. We find that the BS system utility becomes higher when $\eta = 8$ than that when $\eta = 4$ as the number of UEs rises. Because the number of coalitions becomes skyrocketing, achieving the original system objective that facilitates users to share resource with its neighbors as possible as they can rather than compete for cloud's resource. Obviously, the workload of BS is mitigated to maintain the cloudlets. Observe that the U^G when $\eta = 8$ is larger about 9.23% than that when $\eta = 4$ if UEs have increased to 80. On the contrary, the individual utility is rather low when $\eta = 8$ compared with that when $\eta = 4$ as the number of UEs increases. Because more and more UEs participate in competing for cloud's resources, leading to the partition of weight that have been reduced. Furthermore, the sum of weight ratio is fixed to 1. The variable quantity of $F[w]$ in (15) is lower than that of $D[e]$, which is consumed by the sharing procedure. Therefore, the curve of individual utility is still declining. It is worth mentioning that the curves of BS and the individual utility is smoothly decreasing even if the users are increasing, which means that our proposed DARL with the prioritized sampling is compatible with the participation of some uncertain users. Overall, our proposed algorithm possesses the scalability.

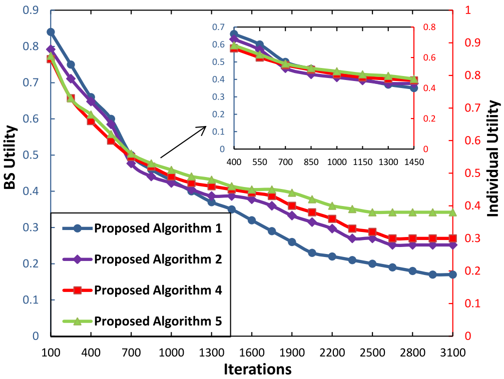

Fig. 6. Comparison of BS utility and individual utility with iterations under different algorithms.

The comparison of BS's utility and the individual utility with iterations under different algorithms is depicted as Fig. 6. From a holistic perspective, since the definition of BS utility in Section III-D, the trend of BS utility is not decreasing until convergence with the growth in iterations, which is in contrast to that of individual utility. Clearly, BS's maintenance cost incurred by managing the users who share idle resource, while it is noted that BS utility of DDPG in Algorithm 1 is higher than that of Algorithm 2 at initial phase. On one hand, the reason is that Algorithm 2 needs to scan the buffer and adopts the prioritized sampling to train the deep model. On the other hand, the efficiency of the analogous DDPG algorithm is high according to (13). There is no need for DDPG to compute the priority that is obtained by the response of users. However, the approach turns to be worse than our improved Algorithm 2 as iterations rise. Aiming to find the stable solution of U^G, the iterations of Algorithm 1 are more than Algorithm 2. In other words, the convergence rate of DARL approach with prioritized sample is faster than the conventional DDPG algorithm. From the views of UEs, It is observed that the convergence of DQN algorithm with coalitional game is better than that without coalition. To illustrate, Algorithm 4 and Algorithm 5 experience about 2500 and 2350 iterations, respectively. In particular, the subgraph shown in Fig. 6 from 400 iterations to around 1500 iterations is deliberately enlarged. It is observed that gradual descent of curves about Algorithm 2 and Algorithm 5 among the special iterations, suggesting that the performance can be guaranteed as the amounts of iterations are rapidly increasing. The turning point that the BS utility of Algorithm 2 turns to higher than that of Algorithm 1 occurs at about 700 iterations. While the individual utility of Algorithm 5 is continuously higher than that of Algorithm 4, showing that the coalition game method with DQN performs better performance than that without the consideration of coalitions. To sum up, it demonstrates that the enhanced algorithms like Algorithms 2 and 5 outperform the counterparts.

VI. Conclusion

In this article, we developed a learning-enabled incentive mechanism with coalitional game for resources sharing to boost the efficient utilization of spare resources in UEs as well as the cloud servers. Since the competition and cooperation are ubiquitous in resource sharing, we managed to address the brand new challenges about incentive mechanism from the view of the pervasive deep learning model and coalition game. To this end, the double actor-critic RL framework for ease of comparison in simulations was first presented. Second, a prioritized sampling with the above-mentioned learning framework was explored, which was compatible with the real-world utility of the individual. Third, the advanced DQN algorithm at UE's side was investigated, which was combined with the designed energy model. Subsequently, the DQN integrated coalition game and the energy model was developed to further promote the IoT devices to make a precise decision. Last but not least, extensive simulations had coincided with the game-theoretical proofs that learning-enabled incentives with coalitional game outperformed the counterparts and could converge to a Nash-stable solution.

In future work, it is necessary to make the data-driven learning model combine with the common model-based formulation, which is worth to be concerned. The reason is that the nature of trial and error for learning model at the initial training phase will cause the oscillation of learning result, especially for some incomplete information or small amount of data offered.

References

[1] S. D'Oro, A. Zappone, S. Palazzo, and M. Lops, "A learning approach for low-complexity optimization of energy efficiency in multicarrier wireless networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 5, pp. 3226–3241, May 2018.

[2] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surveys Tut.*, vol. 19, no. 3, pp. 1628–1656, Jul.–Sep. 2017.

[3] X. Gan, Y. Li, W. Wang, L. Fu, and X. Wang, "Social crowdsourcing to friends: An incentive mechanism for multi-resource sharing," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 3, pp. 795–808, Mar. 2017.

[4] Z. Zhang, D. Zhao, J. Gao, D. Wang, and Y. Dai, "FMRQ-A multiagent reinforcement learning algorithm for fully cooperative tasks," *IEEE Trans. Cybern.*, vol. 47, no. 6, pp. 1367–1379, Jun. 2017.

[5] Y. Li, D. Shi, and T. Chen, "False data injection attacks on networked control systems: A stackelberg game analysis," *IEEE Trans. Autom. Control*, vol. 63, no. 10, pp. 3503–3509, Oct. 2018.

[6] H. Cao and J. Cai, "Distributed multiuser computation offloading for cloudlet-based mobile cloud computing: A game-theoretic machine learning approach," *IEEE Trans. Veh. Technol.*, vol. 67, no. 1, pp. 752–764, Jan. 2018.

[7] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–14.

[8] L. Chen, J. Lingys, K. Chen, and F. Liu, "Auto: Scaling deep reinforcement learning for datacenter-scale automatic traffic optimization," in *Proc. Conf. ACM Special Interest Group Data Commun.*, 2018, pp. 191–205.

[9] A. Sadeghi, F. Sheikholeslami, and G. B. Giannakis, "Optimal and scalable caching for 5G using reinforcement learning of space-time popularities," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 180–190, Feb. 2018.

[10] H. Zhang, Y. Nie, J. Cheng, V. C. M. Leung, and A. Nallanathan, "Sensing time optimization and power control for energy efficient cognitive small cell with imperfect hybrid spectrum sensing," *IEEE Trans. Wireless Commun.*, vol. 16, no. 2, pp. 730–743, Feb. 2017.

[11] H. Wang, T. Huang, X. Liao, H. Abu-Rub, and G. Chen, "Reinforcement learning for constrained energy trading games with incomplete information," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3404–3416, Oct. 2017.

[12] Y. Wang *et al.*, "Multi-resource coordinate scheduling for earth observation in space information networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 2, pp. 268–279, Feb. 2018.

[13] P. V. R. Ferreira *et al.*, "Multiobjective reinforcement learning for cognitive satellite communications using deep neural network ensembles," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 5, pp. 1030–1041, May 2018.

[14] Z. M. Fadlullah *et al.*, "State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2432–2455, Oct.–Dec. 2017.

[15] S. Toyer, F. Trevizan, S. Thiébaux, and L. Xie, "Action schema networks: Generalised policies with deep learning," in *Proc. SIGCOMM*, 2018, pp. 6294–6301.

[16] S. Chinchali *et al.*, "Cellular network traffic scheduling with deep reinforcement learning," in *Proc. Assoc. Advancement Artif. Intell.*, 2018, pp. 766–774.

[17] H. Yu and M. J. Neely, "Learning aided optimization for energy harvesting devices with outdated state information," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, 2018, pp. 1853–1861.

[18] S. Wang *et al.*, "When edge meets learning: Adaptive control for resource-constrained distributed machine learning," in *Proc. IEEE INFOCOM Int. Conf. Comput. Commun.*, 2018, pp. 63–71.

[19] Y. Wei, F. R. Yu, M. Song, and Z. Han, "User scheduling and resource allocation in hetnets with hybrid energy supply: An actor-critic reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 17, no. 1, pp. 680–692, Jan. 2017.

[20] J. Zhu, Y. Song, D. Jiang, and H. Song, "A new deep-q-learning-based transmission scheduling mechanism for the cognitive internet of things," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2375–2385, Aug. 2018.

[21] I. AlQerm and B. Shihada, "Energy-efficient power allocation in multitier 5G networks using enhanced online learning," *IEEE Trans. Veh. Technol.*, vol. 66, no. 12, pp. 11 086–11 097, Dec. 2017.

[22] J. Pan, X. Wang, Y. Cheng, and Q. Yu, "Multisource transfer double DQN based on actor learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2227–2238, Jun. 2018.

[23] I. Grondman, L. Busoniu, G. A. D. Lopes, and R. Babuska, "A survey of actor-critic reinforcement learning: Standard and natural policy gradients," *IEEE Trans. Syst., Man, Cybern. C, Appl.Rev.*, vol. 42, no. 6, pp. 1291–1307, Nov. 2012.

[24] K. G. Vamvoudakis and F. L. Lewis, "Online actor–critic algorithm to solve the continuous-time infinite horizon optimal control problem," *Automatica*, vol. 46, no. 5, pp. 878–888, 2010.

[25] V. Mnih, K. Kavukcuoglu, and D. E. A. Silver, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[26] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proc. 31st Int. Conf. Mach. Learn.*, 2014, pp. 387–395.

[27] Z. Xu *et al.*, "Experience-driven networking: A deep reinforcement learning based approach," in *Proc. IEEE INFOCOM Int. Conf. Comput. Commun.*, 2018, pp. 1871–1879.

[28] G. Rishwaraj, S. G. Ponnambalam, and C. K. Loo, "Heuristics-based trust estimation in multiagent systems using temporal difference learning," *IEEE Trans. Cybern.*, vol. 47, no. 8, pp. 1925–1935, Aug. 2017.

[29] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," 2015, *arXiv:1511.05952*, pp. 1–21.

[30] H. Wu *et al.*, "The tick programmable low-latency SDR system," in *Proc. Int. Conf. Mobile Comput. Netw.*, 2017, pp. 101–114.

[31] L. Zhou, P. Yang, C. Chen, and Y. Gao, "Multiagent reinforcement learning with sparse interactions by negotiation and knowledge transfer," *IEEE Trans. Cybern.*, vol. 47, no. 5, pp. 1238–1250, May 2017.

[32] C. Yang *et al.*, "DISCO: Interference-aware distributed cooperation with incentive mechanism for 5G heterogeneous ultra-dense networks," *IEEE Commun. Mag.*, vol. 56, no. 7, pp. 198–204, Jul. 2018.

[33] B. Zhang, X. Mao, J.-L. Yu, and Z. Han, "Resource allocation for 5G heterogeneous cloud radio access networks with D2D communication: A matching and coalition approach," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 5883–5894, Jul. 2018.

[34] N. Sawyer and D. B. Smith, "A nash stable cross-layer coalitional game for resource utilization in device-to-device communications," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8608–8622, Sep. 2018.

[35] F. Cheng, X. Zhang, C. Zhang, J. Qiu, and L. Zhang, "An adaptive mini-batch stochastic gradient method for AUC maximization," *Neurocomputing*, vol. 318, pp. 137–150, 2018.

[36] T. Wang, L. Song, Z. Han, and B. Jiao, "Dynamic popular content distribution in vehicular networks using coalition formation games," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 9, 1, S, pp. 538–547, Sep. 2013

[37] TensorFlow, [Online]. Available: https://www.tensorflow.org/.

[38] Y. B. I. Goodfellow and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016, [Online]. Available: http://www.deeplearningbook.org/

[39] X. Ma, J. Liu, and H. Jiang, "Resource allocation for heterogeneous applications with device-to-device communication underlaying cellular networks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 1, pp. 15–26, Jan. 2016.

Li Yu received the M.S. degree in computer applications technology from Zhengzhou University, Zhengzhou, China, in 2016. She is currently working toward the Ph.D. degree in computer system architecture with Wuhan University, Wuhan, China.

Her research includes machine learning, resource management, Internet of Things, mobile edge computing, mobile network optimization, and software defined network. She was funded by China Scholarship Council as a Joint Training Ph.D. Student in Simon Fraser University in 2019.

Zongpeng Li (SM'12) received the B.E. degree in computer science from Tsinghua University, Beijing, China, in 1999, and the Ph.D. degree from the University of Toronto, Toronto, ON, Canada, in 2005.

He has been affiliated with University of Calgary and then Wuhan University. His research interests include computer networks, network algorithms, and cloud computing.

Dr. Li received the Outstanding Young Computer Scientist Prize from the Canadian Association of Computer Science, and a few best paper awards from conferences in related fields.

Jiangchuan Liu (F'17) received the B.Eng. degree (*cum laude*) in computer science from Tsinghua University, Beijing, China, in 1999, and the Ph.D. degree in computer science from The Hong Kong University of Science and Technology, Hong Kong, in 2003.

He is currently a University Professor with the School of Computing Science, Simon Fraser University, Burnaby, BC, Canada.

Dr. Liu is the Steering Committee Chair of the IEEE/ACM IWQoS, from 2015 to 2017, and the TPC Co-Chair of the IEEE IC2E 2017 and the IEEE/ACM IWQoS 2014. He serves as an Area Chair for the IEEE INFOCOM, ACM Multimedia, and the IEEE ICME. He has served on the Editorial Boards of the IEEE TRANSACTIONS ON BIG DATA, the IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, the IEEE ACCESS, the IEEE INTERNET OF THINGS JOURNAL, *Computer Communications*, and *Wireless Communications and Mobile Computing* (Wiley).

Ruiting Zhou received the M.S. degree in telecommunications from the Hong Kong University of Science and Technology, Hong Kong, in 2008, the M.S. degree in computer science from the University of Calgary, Calgary, AB, Canada, in 2012, and the Ph.D. degree in 2018 from University of Calgary.

She has been an Associate Professor with the School of Cyber Science and Engineering, Wuhan University, Wuhan, China, since June 2018. Her research interests include cloud computing, machine learning, and mobile network optimization. She has published research papers in top-tier computer science conferences and journals, including IEEE INFOCOM, THE IEEE/ACM TRANSACTIONS ON NETWORKING, IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, and IEEE TRANSACTIONS ON MOBILE COMPUTING.

Dr. Zhou also serves as a Reviewer for journals and international conferences such us THE IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON CLOUD COMPUTING, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON SMART GRID, and IEEE/ACM International Symposium on Quality of Service. She held NSERC Canada Graduate Scholarship, Alberta Innovates Technology Futures Doctoral Scholarship, and Queen Elizabeth ll Graduate Scholarship from 2015 to 2018.