Multiagent RL Aided Task Offloading and Resource Management in Wi-Fi 6 and 5G Coexisting Industrial Wireless Environment

Fanqin Zhou, *Member, IEEE*, Lei Feng, *Member, IEEE*, Michel Kadoch, *Life Senior Member, IEEE*, Peng Yu, *Senior Member, IEEE*, Wenjing Li, *Member, IEEE*, and Zhili Wang, *Member, IEEE*

***Abstract*—With the emergence of industrial Internet of Things (IIoT), intensive computation workload will be imposed to industrial end units (IEUs). By leveraging mobile edge computing (MEC), the local computational tasks can be offloaded to servers deployed in mobile edge networks with low latency. This article proposes the intelligent cost-and-energy-effective task offloading in the 5G and Wi-Fi 6 coexisting heterogeneous IIoT networks. The novel joint task scheduling and resource allocation approach comprises the following two parts: a Lyapunov optimization-based component to decide local task scheduling and computing power and an online multiagent reinforcement learning component together with a game theory-based algorithm to select offloading link and decide transmit power, respectively. Simulation results demonstrate the proposed approach holds obvious advantage over the compared "intuition" and "cost optimal" approaches in the efficiency of making comprehensive decision that improves energy efficiency and cost while controlling task delay in the multi-IEU and multiaccess-node MEC systems.**

***Index Terms*—Industrial Internet of Things (IoT), mobile edge computing (MEC), multiagent learning, resource management, task offloading.**

I. Introduction

IN THE recent wave of industrial digitization, mobile edge computing (MEC) is building the new paradigm by processing the task nearby or directly within the factory [1]. It resolves the issue about latency and security in cloud computing on one hand; on the other hand, it alleviate the local computing resource cost of lightweight industrial end units (IEUs), which are widely planted into industrial system for environment monitoring, data collection, and remote control [2]. However, in the edge computing paradigm, the rich data required in task processing and decision making needs to be transmitted efficiently from IEUs to MEC server. Therefore, it is necessary to deploy wireless technologies for industrial factory since they bring the ubiquitous access for distributed IEUs [3], [4]. It is widely believed that 5G and Wi-Fi 6 are the most promising ones [5] and will go on developing toward high data rate and better quality of service in their technical roadmaps [6].

5G and Wi-Fi 6 use similar physical technologies, such as orthogonal frequency-division multiple access (OFDMA) and multi-input and multi-output (MIMO), but they have different networking conditions. 5G has the advantages over Wi-Fi 6 on the security and quality of service in operators licensed frequency spectrum. However, due to the commercial nature, 5G generally has higher cost for per bit data delivery. Compared to 5G, Wi-Fi 6 access points can be deployed more flexible and Wi-Fi communication is of lower per-bit cost and more energy-efficient, which are more important to the battery-powered wireless devices, such as cruising robots and built-in modules [7]. Thus, coordinating the two types of networks with considerations of instant performance, cost, and energy consumption is essential for MEC task offloading. As a result, the joint optimization of computational task offloading and resource management in Wi-Fi 6 and 5G coexisting industrial wireless environment is the prominent problem that needs to be addressed.

Task offloading in MEC-enabled systems has been investigated in a number of papers. In [8]–[11], mathematical optimization approaches are utilized to derive statistic solutions. However, they generally have difficulty in characterizing the temporal relevance of the optimization variables in adjacent time slots, making the derived solution not optimal in the long term; the large algorithmic complexity also makes these approaches hard to extend to large-scale industrial Internet scenarios. In some papers, such as [12]–[16], machine learning approaches are utilized to address the abovementioned problems. Nevertheless, they are all based on a single-agent setting. For large-scale industrial Internet, it is still not feasible to train centralized agents, since the action space can grow extremely fast with network scale enlarging, which will take up a lot of memory and bring a lot of computing overhead.

Manuscript received May 26, 2020; revised May 13, 2021 and July 19, 2021; accepted August 10, 2021. Date of publication August 24, 2021; date of current version February 2, 2022. This work was supported by Beijing Natural Science Fund-Haidian Original Innovation Joint Fund under Grant L192003. Paper no. TII-20-2629. *(Corresponding authors: Lei Feng; Wenjing Li.)*

Fanqin Zhou, Lei Feng, Peng Yu, Wenjing Li, and Zhili Wang are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: fqzhou2012@bupt.edu.cn; fenglei@bupt.edu.cn; yupeng@bupt.edu.cn; wjli@bupt.edu.cn; zlwang@bupt.edu.cn).

Michel Kadoch is with the Ecole de Technologie Superieure (ETS), University of Quebec, Montreal, QC H3C 1K3, Canada (e-mail: Michel.Kadoch@etsmtl.ca).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2021.3106973.

Digital Object Identifier 10.1109/TII.2021.3106973

Different to the abovementioned literature, this article focuses on cost and energy efficient computing task execution by jointly optimizing the task offloading decision, local CPU clock speed, and selection of network access and the corresponding transmit power of individual IEUs in each time slot. The Lyapunov optimization-based algorithm, we refer to as joint task scheduling and resource allocation (JTSRA), is proposed to decide the local execution tasks and the corresponding CPU clock speed, while an online multiagents reinforcement learning (RL) component together with a game theory-based heuristic algorithm is designed for joint decision of wireless network interface and transmit power of each individual industrial device. Performance evaluations are carried out for the proposed algorithms and the results demonstrate that the proposed approach is able to control the execution delay, power consumption, and cost in the multiuser MEC systems with multinode access selection in the 5G and Wi-Fi heterogeneous network.

The main contribution of this article includes the following. 1) The proposed tasks offloading model is the first, to the best of authors' knowledge, to characterize the emerging industrial task offloading scenarios where 5G and Wi-Fi 6 networks coexist to support multipath offloading. The derived optimization model takes into account optimizing the utilization of Wi-Fi 6 network and computing resources together with the subscribed commercial 5G network and MEC services, as well as the multinode network access selection, from the perspective of energy consumption, cost efficiency, and long-term stability of task queue length and delay. 2) The combination of Lyapunov optimization and multiagent deep reinforcement learning (DRL) is investigated to design the online optimization approach for rapid decisions on task scheduling, offloading link selection, and transmit power of each IEU. Benefiting from the combination, each decision in the proposed approach can be made according to low-complexity methods, which can considerably improve the efficacy and efficiency in solving the joint optimization model and the simulation results validate the execution effectiveness.

The rest of this article is organized as follows. Section II reviews the related work, while Section III introduces the system model. The formulation of the JTSRA problem is depicted in Section IV where the algorithmic framework of JTSRA is also presented. The online multiagent reinforcement learning (MARL) approach is presented in Section V. Performance evaluation and the results are shown in Section VI. Finally, Section VII concludes this article.

II. Related Work

With the rapid development of industrial intelligentization, increasing industrial applications start to call for massive computing resources. Computing tasks offloading from resource-limited IEUs to MEC servers via wireless connections becomes an emerging paradigm to meet the demand for computing resources of industrial applications. 5G and Wi-Fi 6 are seen to be working together in industrial scenarios to provide reliable network connectivity between IEUs and MEC servers [3]. However, the inconsistency between 5G and Wi-Fi systems poses some challenges, one of which is the heterogeneous networks resource management for each IEU, including network node selection and power control for task offloading. In existing research, Liu *et al.* [4] studied the scenarios in which Internet of Things (IoT) devices offload their computing tasks to edge servers through three multiple access transmission methods (hybrid NOMA, pure NOMA, and pure OMA). Qin *et al.* [17] investigated the energy-efficient task offloading problem for large latency-sensitive computing services in MEC-enhanced multiple radio access technologies networks, considering the limit of stringent latency and residual battery energy. However, these previous work did not take the coexistence of 5G and Wi-Fi 6 into account. Since Wi-Fi 6 and 5G are different in the perspectives of energy consumption, cost, and latency [5], [18], [19], how to comprehensively take these factors in the task offloading decision is a critical problem in industrial networks.

The existing research on MEC offloading mainly focused on the situation in a single base station with homogeneous technologies, and usually failed to consider the problem of task offloading in heterogeneous networks. The methods, including Lyapunov optimization, convex optimization, and game theory, are commonly adopted. For example, in [8], the stochastic joint radio and computational resource management was studied for multiuser MEC systems, and a low-complexity online algorithm based on Lyapunov optimization was proposed to achieve asymptotic optimal. Yang *et al.* [9] designed a joint task shunting and data caching model for the MEC and data center coexisting scenarios, and proposed a resource-constrained delay minimization problem, which was solved by the proposed heuristic algorithm based on genetic algorithm and simulated annealing algorithm. Sun *et al.* [10] studied the delay and energy consumption optimized offloading in the 5G MEC system, and an iterative greedy algorithm was proposed to minimize the delay and energy consumption. Zhang *et al.* [11] proposed a new collaborative edge caching architecture, which leveraged MEC resources and 5G networks to enhance edge caching capabilities. In this article, the problem of MEC offloading in 5G and Wi-Fi 6 heterogeneous networks involves binary variables for offloading decision and wireless access node selection and continuous variables for transmission power selection, as well as nonlinear objective functions and constraints. For complex nonlinear problems in actual systems, the aforementioned traditional optimization methods usually require special structures or properties in the problems, and are difficult to be directly popularized due to the high computing complexity and large problem scale. Meanwhile, these methods are snap-oriented. As the environment changes from time to time, they cannot be guaranteed to be the best strategies over a large timescale to serve massive IEUs.

These challenges fall into the research of machine learning. Some common methods such as supervised and unsupervised learning, RL, deep learning, and DRL have been used in MEC [20]. Among them, DRL is especially suitable for decision-making in dynamic resource allocation since it enables a unique paradigm for decision systems to accumulate and utilize experience and to be adaptive to changing environments. Wang *et al.* [12] proposed a deep Q network (DQN)-based resource allocation algorithm to balance resource utilization under varying MEC environment. Wei *et al.* [13] utilized the

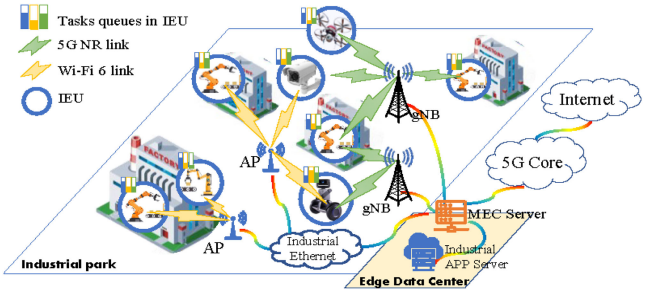

Fig. 1. Mobile edge computation offloading for IIoT system.

actor–critic DRL framework to tackle the issues of content caching strategy, computation offloading policy, and radio resource allocation. However, these research works are all based on a single learning agent, and hence, the number of actions will grow extremely fast with network scale enlarging. Although edge computing has been introduced to the industry, the single agent learning paradigm still faces the problem of great resource and time cost of model training. These limits constrain the practical application of DRL in industrial scenarios. MARL, such as WoLF-PHC, can overcome the explosion of action space by defining each agent with a specific action dimension and implementing each agent in a distributed way based on its local information to reduce the computational cost. Especially, WoLF-PHC provides a distributed learning framework based on global feedback, which is suitable for the scenario where agents cannot communicate frequently. The superiority of WoLF-PHC in complexity is demonstrated in [21] and [22] when solving resource allocation problems. Additionally, the restrained use of intelligent learning algorithms has also attracted attention. In [22], distributed learning is used to learn and predict the key parameters of the channel model, which greatly reduces the complexity of the learning model and system. The abovementioned work inspired us to combine DQN with WoLF-PHC to build Deep-WoLF-PHC framework to reduce the learning model complexity and to be adaptive to our scene.

Different to existing research, this article investigates the joint task offloading and resource allocation problem in Wi-Fi 6 and 5G coexisting industrial wireless environment with energy efficiency, cost, and long-term queuing stability considerations, which is implemented through the combination of online approaches including Lyapunov optimization and multiagent DRL and conventional optimization based on game theory.

III. System Model

We assume the sets of IEUs, gNBs, and Wi-Fi 6 APs are $\mathcal{I}$, $\mathcal{J}$, and $\mathcal{K}$, respectively. The numbers of them are I, J, and K, individually. An illustrative scenario of task offloading in heterogeneous industrial IoT (IIoT) networks is shown in Fig. 1. In the observed industrial park, IEUs can execute tasks locally or offload tasks to the MEC server in the edge data center. And the industrial APP server, capable of generating industrial application instances, makes the immigrated tasks executable in MEC server. For effective wireless connection, each IEU is covered by multiple gNBs and APs, which have wired connections to edge data centers. Each IEU has three types of tasks with different properties, and has to decide whether to process certain tasks locally or migrate them to MEC and which access node (next generation NodeB (gNB) or Wi-Fi access point (AP)) will be chosen if task migration is in need, in order to guarantee the tasks processed in time with minimal energy and cost consumption.

A. Task and Processing Model

The computational tasks can be categorized into the following three types. T_l, T_e, and T_o, represent the types of tasks that can be processed only in local device, in MEC server, and offloadable, respectively.

The workload of computing tasks in this article is measured by bits as in [23]. As tasks may have different computational density, we use ν_l, ν_o, and ν_e (in cycles/bit) to represent the computational density of three types of computational tasks, and usually we have $\nu_l < \nu_o < \nu_e$. In each time slot t, the industrial device will generate a number of bits that need to be calculated, $B(t) = (B_l(t), B_o(t), B_e(t))$ (bits) represents the amount of computing workload for T_l, T_o, and T_e tasks, respectively. The values of $B_x(t), x \in \{l, o, e\}$ in each time slot is in independent identical distributed and $\mathbb{E}[B_l(t)] = \lambda_l, \mathbb{E}[B_o(t)] = \lambda_o, \mathbb{E}[B_e(t)] = \lambda_e$, and they are upper bounded by $B_{l,\max}$, $B_{o,\max}$, and $B_{e,\max}$, respectively.

Let $r_{d,i}(t)$ represent the number of CPU cycles of device i at time slot t, where $i \in \mathcal{I}$ indicates an industrial device, $t = \{0, 1, \ldots\}$ indicates the index of time slot whose length is Δt (in sec). As modern CPU has dynamic voltage and frequency scaling capability, the clock speed can be adjusted discretely within a certain range. Then, the $r_{d,i}(t)$ can be defined as $r_{d,i}(t) \in \{r_1, r_2, r_3, \ldots\ldots, r_{\max}\}$.

We use $\theta_i(t) \in \{0, 1\}$ to denote whether T_o or T_l tasks are scheduled in industrial device i at time slot t. $\theta_i(t) = 1$ indicates the T_o tasks are scheduled, otherwise T_l tasks occupies CPU resources. When T_o tasks are not processed locally, they have the opportunity to be offloaded to MEC. The two types of tasks have to wait for scheduling. We use $\vartheta_i(t) \in \{0, 1\}$ to indicate whether T_o or T_e tasks are offloaded to MEC server during time slot t. T_o tasks are to be offloaded if $\vartheta_i(t) = 1$; otherwise, T_e tasks are offloaded.

B. Wireless Transmission Model

In this article, two types of wireless network interfaces are proposed for task offloading, the commercial 5G cellular wireless network with the licensed spectrum and the industrial wireless network with cognitive unlicensed spectrum. IEUs will choose from either of them for task offloading. We use $\kappa_{i,m}(t) \in \{0, 1\}$ to indicate the chosen network interface of industrial device i during time slot t, where 1 stands for network node $m \in \mathcal{J} \cup \mathcal{K}$. As device i can only be served by one gNB or AP, we have $\sum_{m \in \mathcal{J} \cup \mathcal{K}} \kappa_{i,m} = 1$. During time slot t, the data rate for task offloading is determined by the channel condition and the selected network interface. According to [24], it can be calculated as

$$r_{w,i,m}(t) = w_{i,m}(t) \log_2 (1 + \gamma_{i,m}(t)) - \sqrt{U_i/n} f_Q^{-1}(\varepsilon) \quad (1)$$

where $w_{i,m}(t)$ represents the bandwidth allocated to device i, and $\gamma_{i,m}(t)$ is the signal-to-noise ratio of the uplink received signal during time slot t. U_i is the channel dispersion, which can be expressed by $U_i = 1 - 1/(1+\gamma_{i,m}(t))^2$. In this article, the channel dispersion can be approximated as one, and n is the length of the transmission packet, ε is the transmission error probability. Let $G_{i,m}$ represent the channel gain from IEU i to wireless node $m \in \mathcal{J} \cup \mathcal{K}$, $P_{w,i,m}(t)$ represent the transmit power, and we have $0 \le P_{w,i,m}(t) \le P_{w,i,m}(t)\kappa_{i,m}(t)$. If $m \in \mathcal{J}$

$$\gamma_{i,m}(t) = \frac{G_{i,m}P_{w,i,m}(t)}{\eta\left(\sum_{j\in\mathcal{J}/\{m\}}\sum_{i'\in\mathcal{I}/\{i\}} G_{i',m}P_{w,i',j}(t) + \sigma^2\right)}. \quad (2)$$

Otherwise, if $m \in \mathcal{K}$, IEU i chooses Wi-Fi access, and

$$\gamma_{i,m}(t) = \frac{G_{i,m}P_{w,i,m}(t)}{\eta\left(\sum_{k\in\mathcal{K}/\{m\}}\sum_{i'\in\mathcal{I}/\{i\}} G_{i',m}P_{w,i',k}(t) + \sigma^2\right)}. \quad (3)$$

Here, η_1 and η_2 represent the anti-interference coefficient in 5G and Wi-Fi systems, respectively, $G_{i',m}$ is the channel gain from interfering IEU i' to node m. The data rate of IEU i then can be defined as $r_{w,i}(t) = \sum_{m\in\mathcal{J}\cup\mathcal{K}} \kappa_{i,m}(t) r_{w,i,m}(t)$.

C. Task Queue Model

We use $Q_{l,i}(t)$, $Q_{o,i}(t)$, and $Q_{e,i}(t)$ to represent computational task queues of T_l, T_o, and T_e tasks in device i, and the values of them are the amounts of task bits during time slot t. Their update rules for the next time slot $t+1$ are as follows:

$$Q_{l,i}(t+1) = \left[Q_{l,i}(t) - \frac{(1-\theta(t))r_{d,i}(t)\Delta t}{\nu_l(t)} + B_{l,i}(t)\right]^+ \quad (4)$$

$$Q_{o,i}(t+1) = [Q_{o,i}(t) - \frac{\theta(t)r_d(t)\Delta t}{\nu_o(t)} - \vartheta(t)r_w(\kappa(t),t)\Delta t + B_o(t)]^+ \quad (5)$$

$$Q_{e,i}(t+1) = [Q_{e,i}(t) - (1-\vartheta(t))r_w(\kappa(t),t)\Delta t + B_e(t)]^+ \quad (6)$$

where $[x]^+ = \max(x, 0)$. Equation (4) depicts the queue dynamics for the local tasks. The tasks in the queue in time slot $t+1$ is equal to the task in time slot t plus the new arrival task bits $B_{l,i}(t)$ minus the processed tasks $(1-\theta(t))r_{d,i}(t)\Delta_t/v_l(t)$. Equations (5) and (6) can be similarly understood and we will not explain them one by one.

D. Energy Consumption and Cost Model

Each local device's energy consumption consists of computational processing energy and wireless transmission energy.

For the computational part, let $P_{d,i}(t)$ be the consumed power of the CPU for computing tasks during time slot t, then $P_{d,i}(t)$ can be expressed as $P_{d,i}(t) = \alpha r_{d,i}^x(t) + \beta$, where the exponent x ranges from 2 to 3, and α and β are parameters determined by the CPU model [9], [10]. Then, the consumed energy during the time slot is

$$E_{d,i}(t) = P_{d,i}(t)\Delta t. \quad (7)$$

For the wireless transmission part, when tasks are uploaded through licensed 5G cellular spectrum, the energy consumption of industrial device i is expressed as $\sum_{m\in\mathcal{J}} P_{w,i,m}(t)\Delta t$. When the tasks are uploaded through unlicensed spectrum via Wi-Fi network, the energy consumption includes the energy consumption for scanning available spectrum and data transmission, which can be expressed as $E_s + \sum_{m\in\mathcal{K}} P_{w,i,m}(t)\Delta t$, where E_s represents the consumed energy of scanning process. We neglect the static power consumption of IEU as it does not directly contribute to the wireless transmission. As device i can choose any available wireless access node for task offloading, the consumed energy for wireless transmission during slot t can be written as

$$E_{w,i}(t) = \sum_{m\in\mathcal{J}\cup\mathcal{K}} P_{w,i,m}(t)\Delta t + \sum_{m\in\mathcal{K}} \kappa_{i,k}E_s. \quad (8)$$

As to the overall commercial cost, it includes the cost of the local CPU computation and the cost of wireless transmission to the MEC server. It is assumed that the cost of CPU is proportional to the CPU cycles, and the transmission cost is proportional to the number of bits uploaded. We assume the cost per bit uploaded is ζ (in \$/bit) via cellular interface and no additional cost via industrial cognitive network. For the computational cost, we assume the cost in local CPU is proportional to the energy consumption with a coefficient ξ (in \$/Joule), while the computational cost in MEC is ς (in \$/cycle). Let $\kappa_i(t) = \sum_{j\in\mathcal{J}} \kappa_{i,j}(t)$, the total cost at time slot t can be expressed as

$$C_i(t) = \xi E_{d,i}(t) + \xi E_{w,i}(t) + \vartheta_i(t)\varsigma r_{w,i,m}(t)\Delta t/\nu_o + (1-\vartheta_i(t))\varsigma r_{w,i,m}(t)\Delta t/\nu_e + \kappa_i(t)\zeta r_{w,i,m}(t)\Delta t. \quad (9)$$

IV. Long-Term Intelligent Computation Offloading Scheme

A. Problem Formulation

The goal of our control is slot-by-slot deciding the control parameters of each industrial device i for processing task selection, offloading task selection, access node selection, CPU clock speed, and transmission power, $(\theta_i(t), \vartheta_i(t), \kappa_{i,m}(t), r_{d,i}(t), P_{w,i,m}(t))$, so as to minimize total energy consumption and cost with the total task queues remaining stable. The long-term optimization problem can be defined in per-device perspective as follows. For each industrial device i, we have

$$\min_{(\boldsymbol{\theta},\boldsymbol{\vartheta},\boldsymbol{\kappa},\boldsymbol{r_d},\boldsymbol{P_w})} \lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left\{\frac{E_{d,i}(t)}{r_{d,i}(t)} + \frac{E_{w,i}(t)}{r_{w,i}(t)} + C_i(t)\right\} \quad (10)$$

$$\text{s.t.} \quad \limsup_{t\to\infty} \frac{1}{T}\sum_{\tau=0}^{T-1} \mathbb{E}\left\{Q_{i,l}(\tau) + Q_{i,o}(\tau) + Q_{i,e}(\tau)\right\} < \infty \quad (10a)$$

$$\theta_i(t) + \vartheta_i(t) \le 1 \quad (10b)$$

$$\sum_{m\in\mathcal{J}\cup\mathcal{K}} \kappa_{i,m}(t) \le 1 \quad \forall t. \tag{10c}$$

It should be noted that constraint (10c) makes the per-device optimizations related to each other, and $\kappa_{i,m}$ is independent to the optimization of $r_{d,i}$, which makes it possible to decouple the problems of computing task scheduling and transmission decisions. Therefore, in the sequel, we temporally omit the i index for simplification.

B. Lyapunov Drift Penalty Optimization

The algorithm of this article is based on the Lyapunov drift penalty optimization method. To analyze the stability of total task queues in Section IV-A from individual industrial device perspective, we define the following Lyapunov function:

$$L(t) \triangleq \frac{1}{2}\left\{Q_l(t)^2 + Q_o(t)^2 + Q_e(t)^2\right\}. \tag{11}$$

According to (11), we define the Lyapunov drift function as follows:

$$\Delta(L(t)) \triangleq E\left\{L(t+1) - L(t)|\boldsymbol{Q}(t)\right\}. \tag{12}$$

To ensure the stability of the queue and improve the long-time average energy efficiency and cost of the system, we define a drift penalty function with the long-term energy and cost minimization as its component. The drift penalty function is defined as $\Delta(L(t)) + V\mathbb{E}\{E_d(t)/r_d(t) + E_w(t)/r_w(t) + C(t))|\boldsymbol{Q}(t)\}$. And our objective turns to be

$$\min \Delta(L(t)) + V\mathbb{E}\left\{\frac{E_d(t)}{r_d(t)} + \frac{E_w(t)}{r_w(t)} + C(t)|\boldsymbol{Q}(t)\right\} \tag{13}$$

where V is a parameter for delay and cost tradeoff.

Using the queuing dynamics in (4)–(6) and bounds of workload arrival, we can get the upper bound of (13) as follows:

$$\begin{aligned}
&\Delta(L(t)) + V\mathbb{E}\left\{\frac{E_d(t)}{r_d(t)} + \frac{E_w(t)}{r_w(t)} + C(t)\right\}\\
&\le J + V\mathbb{E}\left\{\frac{E_d(t)}{r_d(t)} + \frac{E_w(t)}{r_w(t)} + C(t)\right\}\\
&- \mathbb{E}\left\{\left(\frac{(1-\theta(t))\,r_d(t)}{\nu_l} - B_l(t)\right)Q_l(t)|\boldsymbol{Q}(t)\right\}\\
&- \mathbb{E}\left\{\left(\frac{\theta(t) r_d(t)}{\nu_o} + \vartheta(t) r_w(t) - B_o(t)\right)Q_o(t)|\boldsymbol{Q}(t)\right\}\\
&- \mathbb{E}\left\{((1-\vartheta(t))r_w(t) - B_e(t))Q_e(t)|\boldsymbol{Q}(t)\right\}.
\end{aligned} \tag{14}$$

As the minimization objective is upper bounded by the right-hand side of the abovementioned inequality, we can minimize the upper bound, i.e., the right-hand side of (14) instead of solving the original objective in (13). As J is constant with regard to the optimization variables, the new optimization goal is formulated as follows:

$$\begin{aligned}
&\min_{(\boldsymbol{\theta},\boldsymbol{\vartheta},\boldsymbol{\kappa},\boldsymbol{r_d},\boldsymbol{P_w})} V\mathbb{E}\left\{\frac{E_d(t)}{r_d(t)} + \frac{E_w(t)}{r_w(t)} + C(t)\right\}\\
&- \mathbb{E}\left\{\left(\frac{(1-\theta(t))\,r_d(t)}{\nu_l} - B_l(t)\right)Q_l(t)|\boldsymbol{Q}(t)\right\}\\
&- \mathbb{E}\left\{\left(\frac{\theta(t) r_d(t)}{\nu_o} + \vartheta(t) r_w(t) - B_o(t)\right)Q_o(t)|\boldsymbol{Q}(t)\right\}\\
&- \mathbb{E}\left\{((1-\vartheta(t))r_w(t) - B_e(t))Q_e(t)|\boldsymbol{Q}(t)\right\}.
\end{aligned} \tag{15}$$

For any mean arrivals $\mathbb{E}[B_l(t)] = \lambda_l$, $\mathbb{E}[B_o(t)] = \lambda_o$, $\mathbb{E}[B_e(t)] = \lambda_e$ within capacity region, there exists a stationary randomized control policy π^*, which selects task scheduling $\theta(t), \vartheta(t)$, CPU speed $r_d(t)$, network access $\kappa(t)$, and transmission power $P_w(t)$ every time slot t, which can be proven using Caratheodorys theorem in the similar way as in [25]. Then, we develop the JTSRA algorithm by finding control variables $(\theta_c(t), \theta_w(t), r_w(t))$ which minimize the right-hand side of (14) every time slot.

C. JTSRA Algorithm

Intuitively, at each slot, the tasks with the heaviest queuing load will be processed with first priority. However, there are options for being processed in the local device or immigrated to MEC, so we should treat the computational tasks differently from the immigration tasks, as the former should be paid more attention to the required CPU cycles, i.e., $Q(t)/\nu(t)$, while the latter calls for more attention to the transmitted bits, i.e., $Q(t)$. Following this principle, the proposed JTSRA algorithm is summarized in Algorithm 1.

As local device can only process T_l and T_o tasks, Q_l will be processed if it has large queuing computational load, i.e., $Q_l/\nu_l \ge Q_o/\nu_o$. Therefore, $\theta(t) = 0$, and Q_o and Q_e will compete for network resources for task bits offloading, where the transmission load will be a key factor. If $Q_e \ge Q_o$, tasks in Q_e will be offloaded to MEC during the current slot, i.e., $\vartheta = 0$; otherwise, tasks in Q_o will be offloaded, which means $\vartheta = 1$. Similarly, we can understand other procedures.

Special attention should be paid to the $Q_o(t) > Q_e(t)$ case when $Q_l(t)/\nu_l(t) < Q_o(t)/\nu_o(t)$. It means T_o has priority to be considered, and we should decide whether to process Q_o locally or to immigrate its task bits to MEC by judging the total energy efficiency and cost, i.e., objective function (15). For the simplification of notations, we let

$$Y_d(t) = \min_{r_d(t)} VE_d(t)\left(\frac{1}{r_d(t)} + \xi\right) - \frac{r_d(t)Q_o(t)}{\nu_o} \tag{16}$$

$$\begin{aligned}
Y_w(t) = &\min_{\kappa(t),P_w(t)} VE_w(t)\left(\frac{1}{r_w(t)} + \xi\right)\\
&- r_w(t)(Q_e(t) - V\varsigma/\nu_e - V\zeta\kappa(t))
\end{aligned} \tag{17}$$

$$Z_d(t) = \min_{r_d(t)}\left(VE_d(t)\left(\frac{1}{r_d(t)} + \xi\right) - \frac{r_d(t)Q_l(t)}{\nu_l}\right. \tag{18}$$

Algorithm 1: JTSRA at each time slot t.

input : Q_l, Q_o, Q_e,
output : $\theta(t), \vartheta(t), P_w(t), \kappa_{(t)}$

if $Q_l(t)/\nu_l(t) \geq Q_o(t)/\nu_o(t)$ **then**
  $\theta(t)^* \leftarrow 0$; // process T_l locally
  $r_d^*(t) \leftarrow \arg_{r_d(t)} Y_d(t)$ in (18)
  if $Q_o(t) \leq Q_e(t)$ **then**
    $\vartheta(t) \leftarrow 0$; // offload T_e
    $(\kappa^*(t), P_w(t)^*) \leftarrow \arg_{(\kappa(t), P_w(t))} Y_w(t)$ in (17)
  else
    $\vartheta(t) \leftarrow 1$; // offload T_o
    $(\kappa^*(t), P_w(t)^*) \leftarrow \arg_{(\kappa(t), P_w(t))} Z_w(t)$ in (19)
else
  if $Q_o(t) \leq Q_e(t)$ **then**
    $\theta \leftarrow 1, \vartheta \leftarrow 0$; // process T_o and offload T_e
    $r_d^*(t) \leftarrow \arg_{r_d(t)} Y_d(t)$ in (16)
    $(\kappa^*(t), P_w(t)^*) \leftarrow \arg_{(\kappa(t), P_w(t))} Y_w(t)$ in (17)
  else
    if $Y_d(t) + Y_w(t) \leq Z_d(t) + Z_w(t)$ **then**
      $\theta \leftarrow 1, \vartheta \leftarrow 0$; // process T_o and offload T_e
      $r_d^*(t) \leftarrow \arg_{r_d(t)} Y_d(t)$ in (16)
      $(\kappa^*(t), P_w(t)^*) \leftarrow \arg_{(\kappa(t), P_w(t))} Y_w(t)$ in (17)
    else
      $\theta \leftarrow 0, \vartheta \leftarrow 1$; // process T_l and offload T_o
      $r_d^*(t) \leftarrow \arg_{r_d(t)} Z_d(t)$ in (18)
      $(\kappa^*(t), P_w(t)^*) \leftarrow \arg_{(\kappa(t), P_w(t))} Z_w(t)$ in (19)

$$Z_w(t) = \min_{\kappa(t), P_w(t)} VE_w(t)\left(\frac{1}{r_w(t)} + \xi\right) - r_w(t)(Q_o(t) - V\varsigma/\nu_o - V\zeta\kappa(t)). \quad (19)$$

Therefore, if $Y_d(t) + Y_w(t) \leq Z_d(t) + Z_w(t)$, locally processing T_o and offloading T_e contribute the smallest value of objective function (15), so $\theta(t) = 1, \vartheta(t) = 0$; otherwise $\theta(t) = 0, \vartheta(t) = 1$. Thus, the scheduling policy can be decided according to the queuing load slot by slot.

Given the scheduling policy, the optimal CPU clock speed at time slot t, $r_d(t)$, can be derived by solving (16) when $\theta = 1$ or solving (18) when $\theta = 0$, while the optimal wireless transmission control parameters $\kappa(t)$ and $P_w(t)$ can be decided by solving (17) when $\vartheta = 0$ or solving (19) when $\vartheta = 1$.

The objective functions (16) and (18) have similar structures, so we only focus on solving (16), which is

$$\min_{r_d(t)} \left\{ V\left(\frac{1}{r_d(t)} + \xi\right)(\alpha r_d(t)^x + \beta)\Delta t - \frac{r_d(t)Q_o(t)}{\nu_o} \right\}. \quad (20)$$

As the second-order derivative function of the objective function is $V(2\beta + \alpha(x-1)r_d(t)^x(x - 2 + x\xi r))/r^3$ and $2 < x < 3$, it is clearly positive. Therefore, its optima is achieved at one of the discrete values of $r_{d,n} \in \{r_{d,1}, \ldots, r_{d,\max}\}$ near the stationary point $r_d^*(t)$ of (20), and bisection search can be performed to find it.

Considering the complexity of solving (17) or (19), and the efficiency requirement on finding a solution to the selection of access node and optimal transmit power for offloading. In the following, an online MARL component together with a game theory-based heuristic algorithm is designed to solve (17). We omit the discussion of (19) due to the similarity.

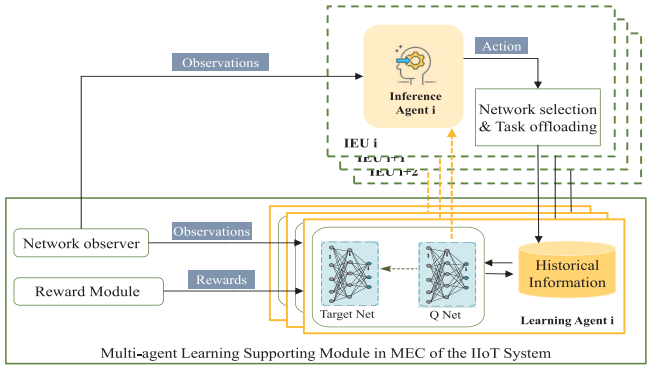

Fig. 2. Framework of the multiagent DRL system.

V. Multiagent DRL-Aided Optimization Approach

This part describes the process of optimizing the energy and cost consumption by deciding the wireless network access choice and the corresponding transmission power through the processed online intelligent learning approach.

Each IEU in the intelligent learning system can be viewed as an inference agent, with a corresponding learning agent on the MEC edge, as depicted in Fig. 2. Inference agents take actions based on its observation and reward in each time slot t according to a learned policy (improved Q-net). However, learning agents are used to approximate value function with deep convolution neural network (CNN) and train learning process with experience replay.

The proposed RL model is depicted as follows.

State: The observed states of the agent i for the corresponding industrial device can be defined as the set $\mathcal{S}_i = \{s_m | (G_{i,m}, \kappa_{i,m})_{1\times 2(J+K)}, \sum_m \kappa_{i,m} \leq 1, m \in \mathcal{J} \cup \mathcal{K}\}$.

Action: The whole action set of agent i is defined as the set $\mathcal{A} = \{a_m | a_m = m, m \in \{1, \ldots, (J+K)\}$. a_m means $\kappa_{i,m} = 1$ and $\kappa_{i,m' \neq m} = 0$.

Policy: π_i is defined as the strategy selection probability of IEU i. The police set $\pi = \{(\pi_m)_{1\times(J+K)}\}$. At the start of each iteration, actions are taken according to the probability police. Based on the strategy probability, we define the transition function as

$$P(s, s,' a) = P(s(t+1) = s' | s(t) = s, a(t) = a). \quad (21)$$

Reward: The function of reward is related to each IEU' actions and states in this system. We consider the overall system energy and cost, but also the fairness of achievable data rate of IEUs in the system in the reward function design. To characterize the fairness of data rate, we adopt the logarithmic utility function $\log(\cdot)$. Instead of utilizing the data rate of other IEUs, in our design, the individual agent only use the total utility F_s in the system, which is defined as $\sum_{m\in\mathcal{J}\cup\mathcal{K}} \sum_{i\in\mathcal{I}} \log(r_{w,i}(t))$, which is available in practical. The final form of the reward function for agent i is as follows:

$$\varrho_i(s, a) = \frac{F_s(t)}{E_{d,i}(t) + E_{w,i}(t) + C_i(t)}. \quad (22)$$

In MARL, the long-term expected reward of state s for agent i is expressed as

$$v_\pi(s) = \sum_{t=0}^{\infty} \mu^t \varrho_i(s,a) \tag{23}$$

where μ represents the discount factor, s' represents the next state s^{t+1}. In Q-learning model, Q function can be defined as

$$Q(s,a) = \varrho(s,a) + \mu \sum_{s'} P(s,s,'a) v(s'). \tag{24}$$

We adopt WoLF-PHC algorithm [26] for its low complexity requirement and strong scalability. In WoLF-PHC, the Q-value update function is defined as

$$\begin{aligned} Q_i^{t+1}(s_i,a_i) &= (1-\alpha^t) Q_i^t(s_i,a_i) \\ &\quad + \alpha^t (r(s_i^t,a_i) + \mu \max_{a'} Q_i(s,'a')) \end{aligned} \tag{25}$$

where the learning rate α^t ranges from 0 to 1, $C(s)$ is used to represent the number of states. Policy π will be recorded and adjusted by applying estimate policy π' according to the following formula:

$$\pi_i'(s,a_i) \leftarrow \pi_i'(s,a_i) + \frac{1}{C(s)} [\pi_i(s,a_i) - \pi_i'(s,a_i)]. \tag{26}$$

We compare the estimate policy π' with π after updating. If $\sum_{a_i \in \mathbf{A}_i} \pi_i(s,a_i) Q_i(s,a_i) > \sum_{a_i \in \mathbf{A}_i} \pi_i'(s,a_i) Q_i(s,a_i)$, π is considered as the winner. Otherwise, π' is considered as the winner. Hyper parameters win-rate δ_w and lose-rate δ_l are intended for winner policy and loser policy, which affect the performance and convergence of the algorithm. If π' is considered to be the winner, δ_w is added to its proportion, and δ_l is reduced to its proportion when π' loses. Besides, we adopt the target network and local network in DQN [27] and optimize the network parameters Θ to minimize the loss function

$$\begin{aligned} \text{Loss}(\Theta) &= \mathbb{E}[(\hat{Q} - Q(s,a;\Theta))^2] \\ \hat{Q} &= \varrho + \gamma \max_{a'} Q(s',a';\Theta) \end{aligned} \tag{27}$$

where $Q(s,a;\Theta)$ is the Q-value calculated by target network.The proposed deep-WoLF-PHC algorithm is summarized in Algorithm 2. In Algorithm 2, considering that the expansion of dimension may cause the inaccuracy and long delay of training distributed DQN, here the state update policy is acquired by WoLF-PHC algorithm following the win-or-lose rule. And due to the reason that on short timescale, we periodically conduct the training, it can be assumed that in each training process, the channel conditions would not change greatly. Besides, after updating the policy, the next state s' is chosen and reward is calculated accordingly. And then we store the experience in the replay memory, which is the biggest difference between our proposed algorithm and conventional WoLF-PHC. Next, the network is updated.

In the multicell and multiuser scenario, the transmission power of each device will interfere with the IEUs, which use the same frequency band in other cells. IEUs have selfish characteristics, they will maximize their transmission power to maximize their interests, ignoring the interference to others. The increase

Algorithm 2: Deep-WoLF Policy Hill-Climbing.

input: user i, win-rate δ_w, lose-rate δ_l, user number I, initial state s, initial action set $\{a_1, ..., a_N\}$, episode number N
output: policy π_i, state s
Initialize: $Q_i(s,a_i) \leftarrow 0$, $\hat{Q}$ update steps C,
Replay memory D_i to capacity M,
Action-value function Q with random weights Θ
Target action-value function $\hat{Q}$ with weights $\hat{\Theta} = \Theta$
for $episode \leftarrow 1 : M$ **do**
  Initialize environment
  $C(s) \leftarrow 0$
  for $j \leftarrow 1 :| \mathcal{A} |$ **do**
    $\pi_i(s,a_j) \leftarrow \frac{1}{|\mathcal{A}|}$, $\pi'(s,a_j) \leftarrow \frac{1}{|\mathcal{A}|}$,
  for $e \leftarrow 1 : N$ **do**
    $a_i(t) \leftarrow \Pi_i(s)$, $\varrho_i \leftarrow \varrho_i(s,a)$, $s' \leftarrow P(s,a_i(t))$
    $Q_i(s,a_i(t)) \leftarrow$
    $Q_i(s,a_i(t)) + \alpha[\varrho_i + \mu \max_{a'} Q_i(s',a') - Q_i(s,a_c)]$
    for $j \leftarrow 1 :| \mathcal{A} |$ **do**
      $C(s) \leftarrow C(s) + 1$
      $\pi_i'(s,a_j) \leftarrow$
      $\pi_i'(s,a_j) + \frac{1}{C(s)}[\pi_i(s,a_j) - \pi_i'(s,a_j)]$
      if $\sum_{a_j \in \mathcal{A}} \pi_i(s,a_j) Q_i(s,a_j) >$
      $\sum_{a_j \in \mathbf{A}_i} \pi_i'(s,a_j) Q_i(s,a_j)$ **then**
        $\sigma \leftarrow \delta_w$
      else
        $\sigma \leftarrow \delta_l$, $\sigma_{s,a} \leftarrow \min(\pi_i(s,a_j), \frac{\sigma}{|\mathcal{A}|-1})$
      if $a_j \neq \arg\max_{a'} Q(s,a')$ **then**
        $\rho_{s,a} \leftarrow -\sigma_{s,a}$
      else
        $\rho_{s,a} \leftarrow \sum_{a' \neq a} \delta_{s,a'}$
      $\pi_i(s,a_j) \leftarrow \pi_i(s,a_j) + \rho_{s,a}$
    $s \leftarrow s'$, get reward ϱ
    Store transition $(s,a,\varrho,s') \rightarrow D$
    Sample minibatch of $(s,a,\varrho,s') \leftarrow D$
    Perform gradient descent on $(\varrho - Q(s,a;\Theta))^2$
    if $C(s) = C$ **then**
      $\hat{Q} \leftarrow Q$

of interference will reduce the SINR and impact the system throughput. We use game theory to analyze the power allocation. The power allocation problem is described as the existence of Nash equilibrium of a noncooperative game. A typical game consists of the following three elements: the participant, the strategy space, and the utility function of the participant. In our case, the participants are IEUs, and the strategy space is the transmission power of each device. The utility function of each participant is designed on the base of (17) and (19) that is $\theta(t) = 1, \vartheta(t) = 0$ and $\theta(t) = 0, \vartheta(t) = 1$. The utility function of device i in the case of $\theta(t) = 1, \vartheta(t) = 0$ can be expressed as

$$\begin{aligned} \Phi_{i,m} &= V \left(\frac{1}{r_{w,i}(t)} + \xi \right) E_w(t) \\ &\quad + r_{w,i}(t)(V\varsigma/\nu_e + V\zeta\kappa_i(t) - Q_e(t)). \end{aligned} \tag{28}$$

Through the choice of transmission power of each IEU, the game process can reach Nash equilibrium point according to the

TABLE I
ENVIRONMENT SIMULATION PARAMETERS

Simulation Parameter [Variable]	Value [Unit]
Total bandwidth of spectrum	200 [MHz]
Number of CPU cycles [r_d]	2~8 $\times 10^8$ [cycles/s]
Local tasks average arrival rate [λ_l]	5×10^4 [bit/s]
Offloadable tasks average arrival rate [λ_o]	1×10^5 [bit/s]
MEC tasks average arrival rate [λ_e]	5×10^4 [bit/s]
Computational density of local tasks [ν_l]	100 [cycles/bit]
Computational density of offloadable tasks [ν_o]	200 [cycles/bit]
Computational density of MEC tasks [ν_e]	240 [cycles/bit]
CPU parameter [α]	0.33
CPU parameter [β]	0.1
CPU parameter [ρ]	1.25 $\times 10^{-25}$
Power optimization parameters [η_1]	0.8
Power optimization parameters [η_2]	1
Energy cost coefficient [ξ]	1.81×10^{-8} [$/J]
Cellular traffic cost coefficient [ζ]	1×10^{-8} [$/bit]
MEC processing cost coefficient [ς]	1.21×10^{-9} [$/cycle]

utility function, and the system reaches the optimal utility, and the interference between devices is the lowest [28].

In order to maximize the utility function, we check the second-order derivative of $\Phi_{i,m}$ with regard to $P_{w,i}$, which is formulated as

$$\frac{G(2GVP - V(2+GP)\log_2(1+GP) + AM^2Q\log_2(1+GP)^3)}{M(1+GP)^2\log(1+GP)^3}$$

where $G = \eta_1(\sum_{j' \in \mathcal{J}/\{j\}} \sum_{i' \in \mathcal{I}/\{i\}} \kappa_{i,'j'}(t) G_{i,'m}(t) P_{w,i,'j'}(t) + \sigma^2)$, $P = P_{w,i}(t)$, $M = w_{i,m}(t)$, $A = Q_{e,i}(t) - V\varsigma/\nu_e - V\zeta\kappa_i(t)$. $\Phi''_{i,m}$ can be proved nonnegative considering the fact that M is numerically much larger than GP. Therefore, in the practical system, the stationary point can be chosen as the optimal solution of the transmit power, and a bisection search can be performed to find the solution. Similar conclusion can also be drawn in the case of $\theta(t) = 0, \vartheta(t) = 1$, and we omit the derivation process to avoid redundancy.

VI. PERFORMANCE EVALUATION

A. Simulation Settings

The simulations are carried out in a scenario with four gNBs and eight APs. We assume the small-scale fading channel power gains are exponentially distributed with unit mean, and the tasks are industrial environment monitoring and data collection. The main parameters are shown in Table I. The proposed approach is compared with two other approaches, denoted as "intuition" and "cost optimal," respectively. The intuition approach utilizes only local information for offloading decisions, while the cost optimal approach performs mathematical optimization when making decisions [25].

These approaches are evaluated from the following aspects:

1) the designed reward showing the trend of convergence and training performance on reward of different learning model hyper parameters;
2) utility and calculation time under different approaches;
3) energy efficiency and cost under different approaches;
4) delay of tasks.

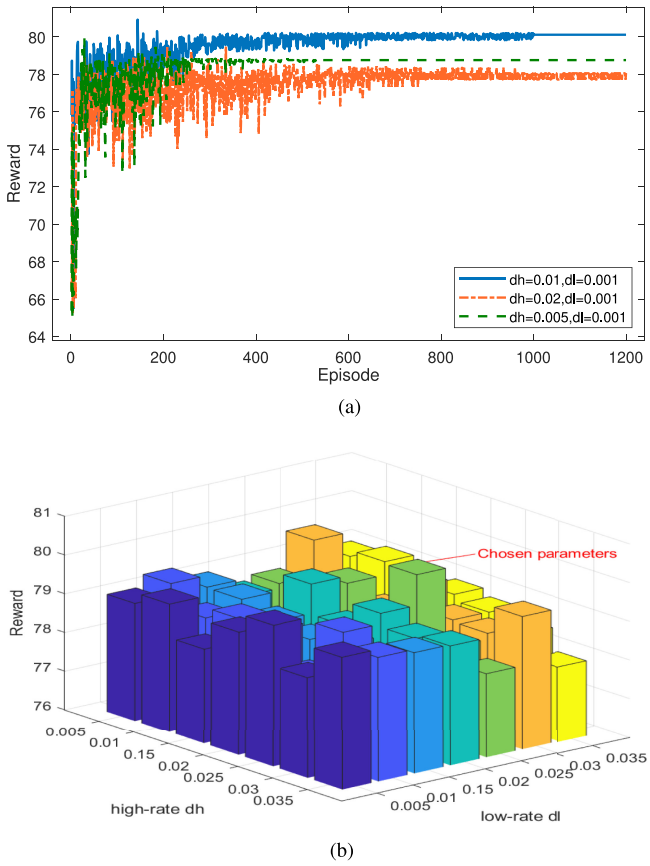

Fig. 3. Offline training simulation results. (a) Convergence under different sets of hyper parameters. (b) Performance on reward with different hyper parameters.

The used deep neural network consists of three fully connected layers including a hidden layer with 64 neurons. The two deep neural networks in the learning agent have the same network structure. The neural networks use the rectified linear unit as the activation function for hidden layers, and the Adam method is adopted for updating the neural network parameters. We set training batch size as 512, memory size as 10 000, and learning rate for Adam optimizer as 0.01. The utilized model and hyper parameters are effective in our simulation, while other effective models like CNN can be used to extend the capability of the learning framework [29].

B. Results and Analyses

We first select a proper set of hyper parameters, dh and dl, for WoLF-PHC algorithm. Fig. 3(a) illustrates the convergence of WoLF-PHC with different sets of hyper parameters. The algorithm converges when hyper parameters change but the final rewards differ. Therefore, we perform a grid search and choose the set of hyper parameters with the largest reward. The results are shown in Fig. 3(b). A range of discrete values of dh and dl are evaluated on the reward. The result indicates that these hyper parameters achieve similar performance. However, when dl ranges from 0.002 to 0.003, reward is relatively higher

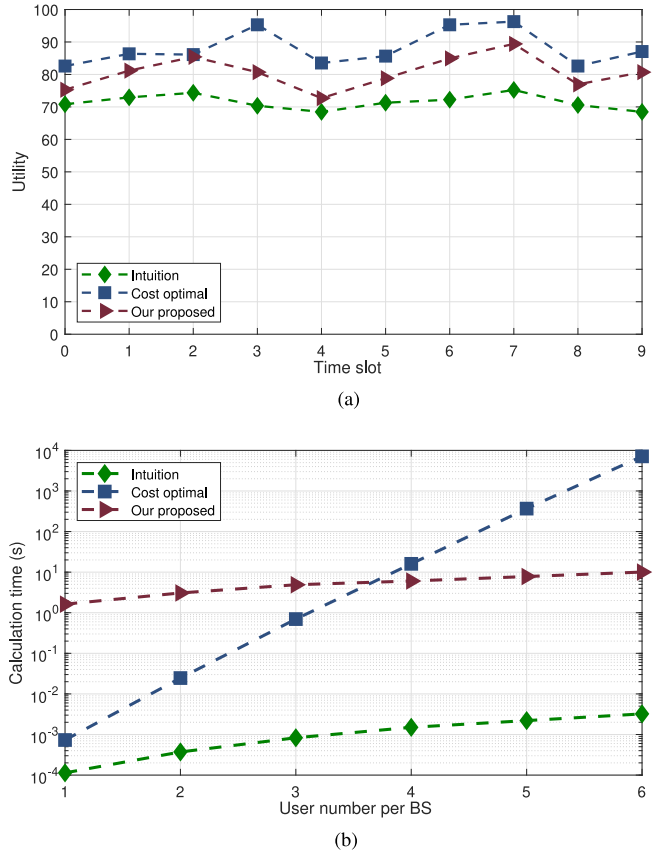

Fig. 4. Online simulation result with the network in continuous time slots. (a) Performance on utility. (b) Calculation time of different approaches.

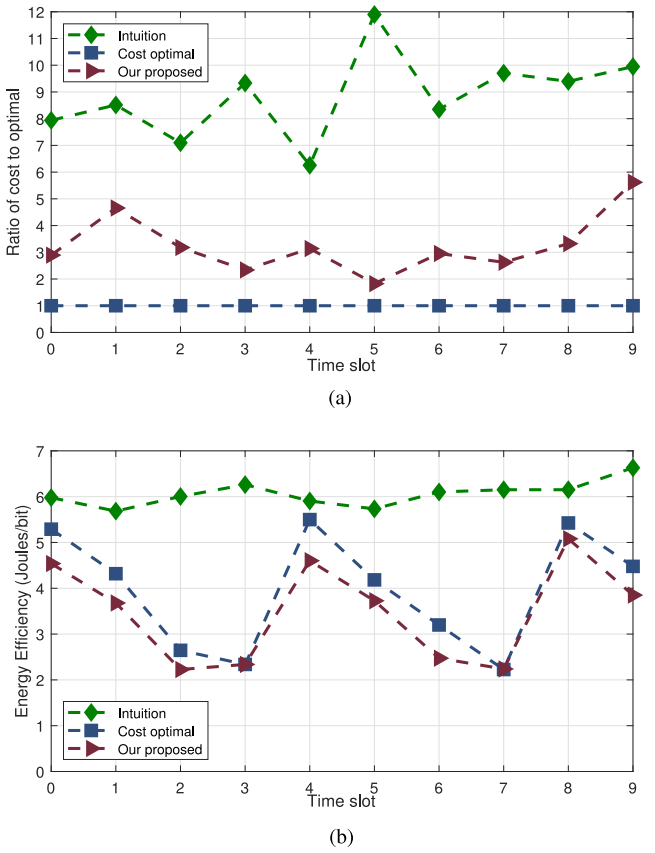

Fig. 5. Comparison of different approaches on cost and power optimization. (a) Performance on ratio of cost. (b) Performance on power optimization.

than others, and we choose 0.025 and 0.0025 for dh and dl, respectively.

In Fig. 4(a), the total value of per IEU utility function is evaluated under three approaches for performance comparison. Under the intuition approach, each device chooses an access node based only on its local information. The proposed approach outperforms the intuition approach and is closer to the cost approach, for it can dynamically choose the optimal scheme considering other actions.

The calculation time of different approaches is shown in Fig. 4(b). Compared with different IEU numbers in each BS, the calculation time grows rapidly when user number increases. When IEU number per BS is up to 6, the calculation time of cost optimal reaches almost 2 h, which is intolerable in our system model. Regarding the intuition approach, although the calculation time maintains little, the performance, such as cost and energy efficiency, is much worse than others.

Fig. 5(a) shows the cost of different approaches in continuous time slots. We set the cost of cost optimal approach as 1 and evaluate the ratio of other approaches' cost compared with that of the cost optimal approach. From the figure, our proposed approach takes the cost about 3–4 times larger than the cost optimal approach; the intuition approach costs almost ten times larger than the cost optimal approach. This is because the cost optimal approach exhaustively searches all the possible network access options and chooses the best one among them, while the intuition approach chooses the network access option, which is only beneficial to itself. In our proposed approach, the user tends to apply the scheme, which is beneficial to the whole system after enough training.

Fig. 5(b) presents the energy efficiency performance of our proposed approach compared with cost optimal and intuition approaches. It should be noted that the lower value of the energy efficiency (Joules/bit) in the figure represents better energy efficiency. Both the computing power and wireless transmission power are considered in the energy efficiency evaluation. Apparently, the consumed energy per bit under intuition approach is steady but much larger than other approaches, because each device chooses its best access node only based on local information. Under the cost optimal approach, the transmission power is minimized but the computing power may increases when the transmission power decreases. Under our proposed approach, the sum of calculation power and transmission power is optimized, which makes the proposed approach outperform the other two on the energy efficiency performance.

Fig. 6(a) depicts the average delay of processing the three types of tasks in each of a series of 10 time slots. We can observe that the proposed approach outperforms the intuition approach and the cost optimal approach by producing the lowest average delay in processing the T_e and T_o tasks, while

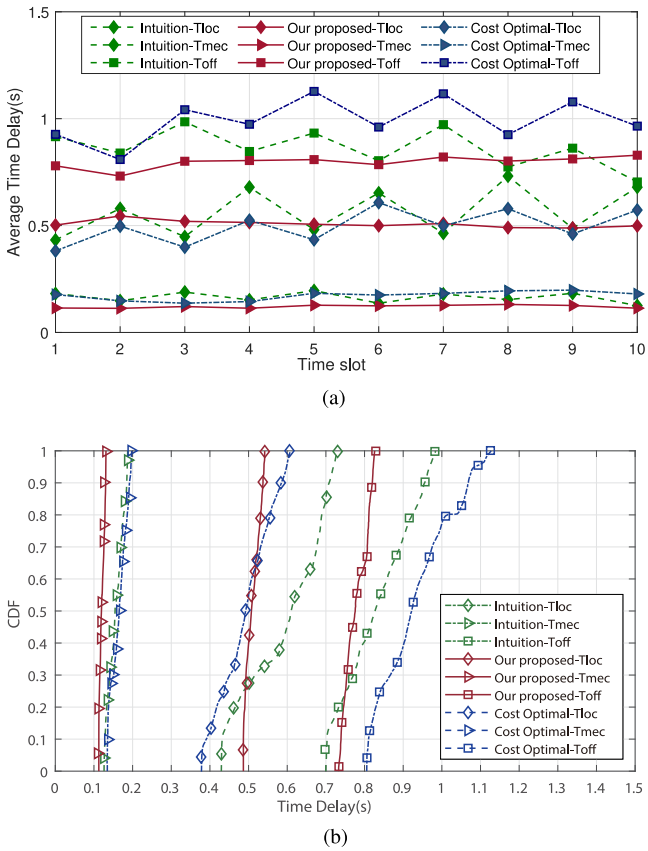

Fig. 6. Simulation results of latency and reliability. (a) Average time delay of different approaches. (b) CDF of average time delay.

having a similar average delay with the other two approaches in processing the T_l tasks. It is possibly because the proposed approach will take advantage of multiple access to select optimal network links for effective task offloading. Due to more tasks can be processed through MEC, more local resource can be saved for T_o and T_l tasks, resulting faster processing speed and lower average latency. Owing to the sufficient computing in MEC, tasks processed in MEC have smallest averaged delay; however, the average delay of the tasks processed locally is relatively large because of the limited computing capacity of IoT devices. The overall packet size of task offloading is larger than others, and it needs to compete with T_e and T_l tasks in the process of occupying local computing resource or communication resource, which leads to the maximum delay of T_o tasks.

To analyze the per task delay performance in case of possible excessive delay of any individual tasks, the fitted cumulative distribution functions (CDFs) of the average time delay of all the tasks in the simulated slots are plotted in Fig. 6(b). The solid curves represented the CDFs of different types of tasks under the proposed approach. From the figure, it is apparent that, for a given threshold of time delay, like 0.15 s, the proposed approach has more robust control of the delay of T_e tasks than the compared approaches. For a given delay threshold, like 0.55 s, T_l tasks can be finished under the proposed approach, while only about 85% and 45% T_l tasks can be finished under the intuition approach and cost optimal approach, respectively. For a given delay threshold, like 0.85 s, T_o tasks can be finished under the proposed approach, while only about 55% and 28% T_o tasks can be finished within the threshold under the intuition approach and cost optimal approach, respectively. From the results, we can see the proposed approach has the advantages in the control of the processing delay of different tasks within a narrow interval, which can further reduce possibility of unfinished tasks in the 5G and Wi-Fi 6 coexisting industrial environment and improve the overall reliability of the MEC system.

From abovementioned results and analyses, we can say the proposed approach has advantages over the compared approaches in achieving cost and energy efficiency by enabling each IEU to make task offloading decisions, intelligently select access nodes, and optimally decide the transmit power, while controlling the delay of tasks in queue to maintain the stability and efficiency of the system.

VII. Conclusion

This article proposed an online approach for each industrial terminal to jointly optimize the computational and communication resource allocation in a bid to reduce the overall power consumption and communication cost in an industrial network scenario. By combining the Lyapunov optimization method, the MARL framework, and the game theory-based heuristic algorithm, the proposed approach is able to decide the computation offloading, wireless access node selection, and transmit power of each IEU in a distributive paradigm for the system overall optimized power consumption and cost. The proposed approach can considerably reduce the complexity of deriving good solution for network scenarios with a large number of terminals with multiple controlling variables. Simulation results validate the effectiveness of our approach.

References

[1] X. Pan, A. Jiang, and H. Wang, "Edge-cloud computing application, architecture, and challenges in ubiquitous power Internet of Things demand response," *J. Renewable Sustain. Energy*, vol. 12, no. 6, 2020, Art. no. 062702.

[2] Y. Chen, Z. Liu, Y. Zhang, Y. Wu, X. Chen, and L. Zhao, "Deep reinforcement learning-based dynamic resource management for mobile edge computing in industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 17, no. 7, pp. 4925–4934, Jul. 2021.

[3] S. Vitturi, C. Zunino, and T. Sauter, "Industrial communication systems and their future challenges: Next-generation ethernet, IIoT, and 5 G," *Proc. IEEE*, vol. 107, no. 6, pp. 944–961, Jun. 2019.

[4] L. Liu, B. Sun, Y. Wu, and D. H. K. Tsang, "Latency optimization for computation offloading with hybrid NOMA-OMA transmission," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6677–6691, Apr. 2021.

[5] E. J. Oughton *et al.*, "Revisiting wireless Internet connectivity: 5G vs Wi-Fi 6," *Telecommun. Policy*, vol. 45, no. 5, 2021, Art. no. 102127. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S030859612100032X

[6] N. Aljeri and A. Boukerche, "Smart and green mobility management for 5G enabled vehicular networks," *Trans. Emerg. Telecommun. Technol.*, vol. 2020, pp. 1–18, May 2020.

[7] A. Zreikat, "Performance evaluation of 5G/WiFi-6 coexistence," *Int. J. Circuits, Syst. Signal Process.*, vol. 14, pp. 904–913, Dec. 2020.

[8] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief, "Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 5994–6009, Sep. 2017.

[9] H. Wang, R. Li, L. Fan, and H. Zhang, "Joint computation offloading and data caching with delay optimization in mobile-edge computing systems," in *Proc. 9th Int. Conf. Wireless Commun. Signal Process.*, 2017, pp. 1–6.

[10] Y. Sun, Z. Hao, and Y. Zhang, "An efficient offloading scheme for MEC system considering delay and energy consumption," *J. Phys.: Conf. Ser.*, vol. 960, no. 1, 2018, Art. no. 012002.

[11] K. Zhang, S. Leng, Y. He, S. Maharjan, and Y. Zhang, "Cooperative content caching in 5G networks with mobile edge computing," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 80–87, Jun. 2018.

[12] J. Wang, L. Zhao, J. Liu, and N. Kato, "Smart resource allocation for mobile edge computing: A deep reinforcement learning approach," *IEEE Trans. Emerg. Topics Comput.*, to be published, doi: 10.1109/TETC.2019.2902661.

[13] Y. Wei, F. R. Yu, M. Song, and Z. Han, "Joint optimization of caching, computing, and radio resources for fog-enabled IoT using natural actor-critic deep reinforcement learning," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2061–2073, Apr. 2019.

[14] S. Wang, M. Chen, X. Liu, and C. Yin, "Task and resource allocation in mobile edge computing: An improved reinforcement learning approach," in *Proc. IEEE Globecom Workshops*, 2019, pp. 1–6.

[15] C. You, K. Huang, H. Chae, and B. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1397–1411, Mar. 2017.

[16] Y. He, F. R. Yu, N. Zhao, and H. Yin, "Secure social networks in 5 G systems with mobile edge computing, caching, and device-to-device communications," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 103–109, Jun. 2018.

[17] M. Qin *et al.*, "Service-oriented energy-latency tradeoff for IoT task partial offloading in MEC-enhanced multi-RAT networks," *IEEE Internet Things J.*, vol. 8, no. 3, pp. 1896–1907, Feb. 2021.

[18] E. J. Oughton *et al.*, "Revisiting wireless Internet connectivity: 5G vs Wi-Fi 6," *Telecommun. Policy*, vol. 45, no. 5, 2021, Art. no. 102127.

[19] M. Qin *et al.*, "Service-oriented energy-latency tradeoff for IoT task partial offloading in MEC-enhanced multi-RAT networks," *IEEE Internet Things J.*, vol. 8, no. 3, pp. 1896–1907, Feb. 2021.

[20] B. Cao, L. Zhang, Y. Li, D. Feng, and W. Cao, "Intelligent offloading in multi-access edge computing: A state-of-the-art review and framework," *IEEE Commun. Mag.*, vol. 57, no. 3, pp. 56–62, Mar. 2019.

[21] Y. Guo and M. Xiang, "Multi-agent reinforcement learning based energy efficiency optimization in NB-IoT networks," in *Proc. IEEE Globecom Workshops*, 2019, pp. 1–6.

[22] T. Mai, H. Yao, F. Li, X. Xu, Y. Jing, and Z. Ji, "Computing resource allocation in LEO satellites system: A stackelberg game approach," in *Proc. 15th Int. Wireless Commun. Mobile Comput. Conf.*, 2019, pp. 919–924.

[23] J. Kwak, Y. Kim, J. Lee, and S. Chong, "DREAM: Dynamic resource and task allocation for energy minimization in mobile cloud systems," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 12, pp. 2510–2523, Dec. 2015.

[24] H. Shariatmadari *et al.*, "Resource allocations for ultra-reliable low-latency communications," *Int. J. Wireless Inf. Netw.*, vol. 24, no. 3, pp. 317–327, Sep. 2017. [Online]. Available: http://link.springer.com/10.1007/s10776-017-0360-5

[25] J. Kwak, O. Choi, S. Chong, and P. Mohapatra, "Processor-network speed scaling for energy-delay tradeoff in smartphone applications," *IEEE/ACM Trans. Netw.*, vol. 24, no. 3, pp. 1647–1660, Jun. 2016.

[26] P. Cook, "Limitations and extensions of the WoLF-PHC algorithm," Thesis Dissertations, Sep. 2007. [Online]. Available: https://scholarsarchive.byu.edu/etd/1222a

[27] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[28] K. Kumar *et al.*, "A survey of computation offloading for mobile systems," *Mobile Netw. Appl.*, vol. 18, no. 1, pp. 129–140.

[29] L. Huang, S. Bi, and Y.-J. A. Zhang, "Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks," *IEEE Trans. Mobile Comput.*, vol. 19, no. 11, pp. 2581–2593, Nov. 2020.