Latency and Energy Minimization in NOMA-Assisted MEC Network: A Federated Deep Reinforcement Learning Approach

Arian Ahmadi, Anders Høst-Madsen and Zixiang Xiong

***Abstract*—Multi-access edge computing (MEC) is seen as a vital component of forthcoming 6G wireless networks, aiming to support emerging applications that demand high service reliability and low latency. However, ensuring the ultra-reliable and low-latency performance of MEC networks poses a significant challenge due to uncertainties associated with wireless links, constraints imposed by communication and computing resources, and the dynamic nature of network traffic. Enabling ultra-reliable and low-latency MEC mandates efficient load balancing jointly with resource allocation. In this paper, we investigate the joint optimization problem of offloading decisions, computation and communication resource allocation to minimize the expected weighted sum of delivery latency and energy consumption in a non-orthogonal multiple access (NOMA)-assisted MEC network. Given the formulated problem is a mixed-integer non-linear programming (MINLP), a new multi-agent federated deep reinforcement learning (FDRL) solution based on double deep Q-network (DDQN) is developed to efficiently optimize the offloading strategies across the MEC network while accelerating the learning process of the Internet-of-Thing (IoT) devices. Simulation results show that the proposed FDRL scheme can effectively reduce the weighted sum of delivery latency and energy consumption of IoT devices in the MEC network and outperform the baseline approaches.**

I. Introduction

The 6G wireless cellular network must support a wide range of new applications with ultra-high reliability and low latency service requirements [1], [2]. Among these emerging applications include, connected and autonomous vehicles (CAVs), extended reality (XR), Internet-of-Things (IoT) networks with strict quality-of-service (QoS) requirements on the end-to-end (E2E) latency (e.g., 1 ms) and reliability (e.g., 10^{-8} packet loss probability) [3]. Moreover, given the constrained computational power and limited battery capacity of IoT devices, meeting predefined task deadlines that require significant resources often becomes unfeasible for these devices. To meet such stringent service requirements, multi-access edge computing (MEC) is an attractive solution to significantly reduce service latency by enabling base stations (BSs) to process computing tasks (e.g., XR rendering) for user equipment (UE), e.g., IoT devices, directly within the radio access network (RAN) without relying on remote cloud servers [4], [5].

While promising, ensuring a consistent performance over a MEC network is challenging due to random wireless channel variations, stochastic task arrival, as well as heterogeneity of edge computing servers and computing tasks. Additionally, considering the constrained resources of IoT devices and edge servers, such as limited bandwidth and computing capabilities, the MEC-supported network is susceptible to becoming overwhelmed, resulting in dropped computing tasks and poor QoS. Hence, it is imperative to develop innovative solutions that jointly optimize the offloading decisions and efficiently allocate edge computing resources for processing tasks from IoT devices within the MEC framework.

Recently, the majority of existing research [4]–[9] has tackled the challenge of joint decision-making on offloading, communication, and allocation of computational resources in ultra-reliable and low-latency MEC (URLL MEC) utilizing methodologies from the field of optimization theory. The power allocation for delivery latency reduction in a device-to-device (D2D) enabled MEC scenario is studied in [9]. In fact, the authors propose a new power allocation algorithm to minimize the total latency while guaranteeing the task computing deadline. However, these algorithms have commonly suffered from issues of scalability, time inefficiency, and high computational overhead. Instead of relying on model-based methods, model-free approaches that leverage machine learning (ML) in combination with the computing capabilities of UEs and BSs can provide new opportunities for enabling URLL MEC.

In this regard, the body of work in [10]–[13] presents several new schemes based on deep neural networks (DNNs) and deep reinforcement learning (DRL) to optimize the network performance for URLL MEC applications. The authors in [11] propose a resource allocation technique using multi-agent DRL in a vehicle-to-vehicle communication network. In [12], the authors present a DRL-based computation and communication resource allocation scheme in a MEC railway IoT network. In [14], the authors develop a joint task, spectrum and transmit power allocation method based on multi-stack RL to minimize the computational and transmission latency in a MEC network. However, a significant drawback of many DRL algorithms is their centralization, leading to scalability issues as the number of devices increases. Additionally, finding an optimal policy becomes computationally complex, especially with the exponential growth of the state and action spaces. Moreover, centralized learning demands IoT devices to share their data to train a global model, potentially compromising privacy.

Recently, distributed learning algorithms that enable users to collaboratively build a unified learning model while conducting local training [15] are developed. One of the most promising distributed learning frameworks is

This research was supported under Grants EECS-1923751 and EECS-1923803.

A. Ahmadi and A. Høst-Madsen are with the Department of Electrical Engineering, University of Hawaii at Manoa, Honolulu, HI 96822 USA (Email: arian.ahmadi72@gmail.com; ahm@hawaii.edu).

Z. Xiong is with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843 USA (Email: zx@ece.tamu.edu).

federated learning (FL) which preserves the data privacy by avoiding data uploading to the parameter server (PS) [16], [17]. In FL scheme, each device utilizes its individual local dataset for training and subsequently transfers its respective local model to the PS for global aggregation. In particular, FL improves collaboration among agents and enhances the scalability of network resource management algorithms. In this regard, the problem of joint power allocation and resource allocation for URLL communications in vehicular networks is investigated in [18]. The authors propose a novel distributed scheme based on FL in order to estimate the tail distribution of the queue lengths. However, the constraint of energy consumption of each user in the network is not considered.

More recently, several research studies have delved into the challenge of reducing latency and energy consumption for IoT devices within the context of a FL system [19]–[21]. For instance, in [19], the authors investigate the problem of jointly optimizing resource and learning performance to reduce communication costs and improve learning performance in wireless FL systems. The authors in [21] focus on optimizing various aspects of FL, including weight compression, convergence analysis and iteration reduction over IoT networks. However, none of the mentioned research works applied FL to enhance the effectiveness in solving a real-world wireless resource allocation problem. In [22], the authors employ FL to facilitate the learning process in DRL, wherein individual local DRL models were trained and subsequently merged collaboratively to construct a comprehensive global DRL model. However, the authors modeled the network as a queuing system without explicitly ensuring QoS for users' tasks and allocation of computation resources.

The main contribution of this paper is a novel framework to optimize the reliability of the MEC IoT network by minimizing joint delivery latency and energy consumption of IoT devices using federated DRL (FDRL). In particular, we design a non-orthogonal multiple access (NOMA) model of heterogeneous network with multiple BSs and multiple IoT devices to dynamically optimize offloading decisions, base station assignment, and channel resource allocation in a decentralized manner across the RAN, while accounting for the constraints of the wireless network and edge computing servers. Given that the proposed problem is a mixed integer non-linear programming (MINLP) and difficult to solve, a new algorithm is developed to jointly solve the offloading decision problem, computing resource and transmit power allocation across the MEC network. Mainly, we first reformulate our problem as a multi-agent DRL problem and solve it using double deep Q-network (DDQN). To enhance both the quality and speed of learning of the proposed algorithm, we integrate FL at the end of each episode. Utilizing FL results in a framework that is both privacy-preserving and scalable, establishing a context for cooperation among agents. Extensive simulation results are carried out to show the superiority of the proposed algorithm through comparisons with those existing schemes. Results indicate that the proposed scheme considerably outperforms the benchmarks under multiple performance metrics.

The rest of the paper is organized as follows. Section II presents the system model. Section III and IV describe the problem formulation and the proposed solution. Simulation results are provided in Section V and conclusions are presented in Section VI.

II. System Model

We consider an MEC heterogeneous network based on NOMA consisting of a set $\mathcal{M}$ of M IoT devices with limited computation and energy resources and a set $\mathcal{N}$ of N BSs as shown in Fig. 1. At each time t, device m needs to process one of the tasks in its queue and can either execute its task locally or offload it to a nearby BS n. Specifically, we use $x_{m,n} \in \{0,1\}$ to indicate the IoT device offloading decisions. When $x_{m,n} = 0$, represents the IoT device uses local processing, Otherwise, the it uploads to BS for processing. Meanwhile, NOMA transmission scheme applies successive interference cancellation (SIC) receivers at the receiving end to realize multi-user detection. Next, we explain the transmission and computing latencies in details.

A. Over-the-Air Transmission Latency and Energy Consumption

If the IoT device m decides to offload its task, it should first transmit it to the BS through wireless channels. Due to the wireless fading channel, some packets may not be decoded successfully at the BS, hence, re-transmission is needed. With this in mind, the transmission latency is given by:

$$\tau_{m,n}^{air} = \sum_{j=1}^{J} \left\lceil \frac{I_m}{r_{m,n,j}} \right\rceil, \tag{1}$$

where $\lceil . \rceil$ is the ceiling function, $J-1$ is the number of re-transmissions, and I_m is the size of the task (in bits). In (1), $r_{m,n,j}$ represents the transmission data rate for the j-th transmission is calculated as follows:

$$r_{m,n,j} = \omega \log_2 \left(1 + \gamma_{m,n,j}\right), \tag{2}$$

where ω is the channel bandwidth. Moreover, $\gamma_{m,n,j}$ is the signal-to-interference-plus-noise-ratio (SINR) and is given by

$$\gamma_{m,n,j} = \frac{G_m G_n P_m h_{m,n,j} L_{m,n}}{\sum_{m' \neq m} P_{m',n} + \sigma_n^2}, \tag{3}$$

where P_m, $P_{m',n}$, and σ_n^2 denote, respectively, the transmit power of IoT device m, the received power from an interfering IoT device m', and the noise power. G_m and G_n are the antenna gains for IoT device m and BS n, respectively. In addition, $h_{m,n,j}$, and $L_{m,n}$ represent, respectively, the Rayleigh fading channel gain for the j-th transmission and path loss between IoT device m and BS n. The channel gain $h_{m,n,j}$ is considered flat-fading over the bandwidth ω and constant during the transmission of one packet.

In addition, energy consumption of device m, while offloading its task to the BS n is given as follows

$$\begin{aligned} E_{m,n}^{air} &= \tau_{m,n}^{air} P_m \\ &= \frac{I_m P_m}{\omega \log_2 \left(1 + \frac{G_m G_n P_m h_{m,n,j} L_{m,n}}{\sum_{m' \neq m} P_{m',n} + \sigma_n^2}\right)}. \end{aligned} \tag{4}$$

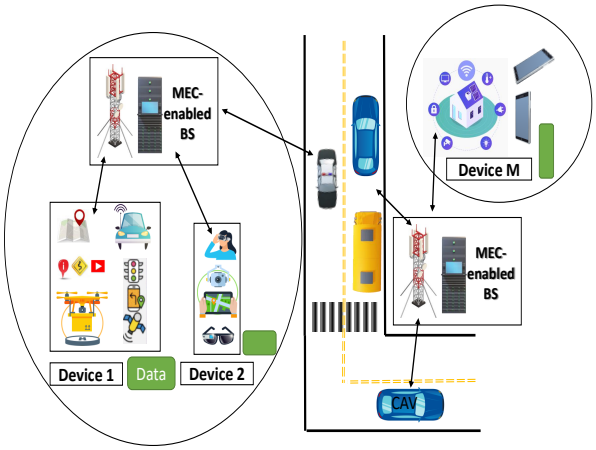

Fig. 1. System Model.

B. Computing Latency at Edge Computing Servers

Computing latency refers to the time needed for executing a task at an edge computing server within the MEC network. The execution time of a task depends on the data to be processed and the tasks submitted, therefore, it can be modeled as a random variable [23]. For instance, the execution time for performing the object detection highly depends on the quality or level of details in the captured images, as well as the type (GPU vs. CPU) and processing resources (e.g., processing bandwidth) of the edge server.

Let f_{max} denote the maximum computing-cycle frequency of edge processors for each BS. If device m offloads its task to the edge server the computation latency would be $\tau_{m,n}^{comp} = \frac{c_m}{f_{m,n}^{edge}}$, where c_m and $f_{m,n}^{edge}$ denote the CPU cycle requirement of the task and average computation capacity of edge server, respectively. From the standpoint of an IoT device, the energy expended to process a task when offloaded to an edge server includes the energy used for task transfer. Hence, the overall energy utilization for edge computing would be equal to $E_{m,n}^{air}$.

In addition, if device m decides to process its task locally, the latency and energy consumption for local computation would vary based on the computation resources allocated for task processing at time t, denoted by f_m^{loc}. Therefore, the local latency and energy consumption for the device are modeled as follows:

$$\begin{aligned} \tau_m^{loc} &= \frac{c_m}{f_m^{loc}}, \\ E_m^{loc} &= \zeta_m \left(f_m^{loc}\right)^2, \end{aligned} \tag{5}$$

where ζ is a constant coefficient which depends on the chip architecture in devices. It's important to note that higher resource utilization, whether in terms of transmit power or computation capacity, reduces the delivery latency at the cost of higher energy consumption. Hence, managing this trade-off requires careful handling through efficient offloading decision-making and precise optimization of both local computation and offloading strategies.

III. MULTI-OBJECTIVE PROBLEM FORMULATION

The main goal of this paper is to minimize the latency of completing task and energy consumption at the same time by jointly accounting the offloading decisions, BS selection and channel resource allocation made by IoT devices. The long-term expected weighted sum of latency and energy consumption between each IoT device m and BS n, is formulated as follows:

$$\tau_{m,n}\left(\mathbf{P}, \mathbf{f}^{loc}, \mathbf{f}^{edge}, \mathbf{x}\right) = \mathbb{E}\left[\lim_{t\to\infty} \frac{1}{t} \sum_{i=0}^{t} (1 - x_{m,n,i})\tau_{m,i}^{loc} + x_{m,n,i}(\tau_{m,n,i}^{comp} + \tau_{m,n,i}^{air})\right], \tag{6}$$

and

$$E_{m,n}\left(\mathbf{P}, \mathbf{f}^{loc}, \mathbf{f}^{edge}, \mathbf{x}\right) = \mathbb{E}\left[\lim_{t\to\infty} \frac{1}{t} \sum_{i=0}^{t} (1 - x_{m,n,i})E_{m,i}^{loc} + x_{m,n,i}E_{m,n,i}^{air}\right], \tag{7}$$

where $\mathbf{P}$, $\mathbf{f}^{loc}$, $\mathbf{f}^{edge}$ and $\mathbf{x}$ represent the vectors of transmit powers, local computing resource allocation, edge server computation resource allocation, local computing and edge offloading decision, respectively.

For any given device m at time t, we model the decision-making problem (DMP), expressed as:

$$\begin{aligned} &\min_{\mathbf{P},\mathbf{f}^{loc},\mathbf{f}^{edge},\mathbf{x}} U = \tau_{m,n} + \lambda E_{m,n} \\ &\text{subject to:} \\ &\text{C1}: (1 - x_{m,n})E_m^{loc} + x_{m,n}E_{m,n}^{air} \le E_{\max}, \\ &\text{C2}: (1 - x_{m,n})\tau_m^{loc} + x_{m,n}(\tau_{m,n}^{air} + \tau_{m,n}^{comp}) \le \tau_{\text{th}}, \\ &\text{C3}: f_m^{loc} \le F_{\max}, \\ &\text{C4}: \sum_{m\in\mathcal{M}} f_{m,n}^{edge} \le f_{\max}, \\ &\text{C5}: x_{m,n} \in \{0, 1\}, \qquad \forall m \in \mathcal{M}, \forall n \in \mathcal{N}, \end{aligned} \tag{8}$$

where λ is a weighting factor. Constraint C1 represents the restriction on the energy resource of the device and that energy utilization should not exceed E_{max}. Constraint C2 expresses that total time for processing the task must meet the maximum latency threshold determined by QoS requirement regardless of whether it is following local processing or offloading to BS processing. Constraints C3 and C4 indicate the local computation capacity with the maximum threshold F_{max} and the sum of the computation frequency allocated to IoT devices should not exceed the maximum computation frequency of the MEC server $f_{\max}$. The constraints C5 indicates that the decision variables of the problem are all binary indicators.

The proposed optimization problem in (8) is a MINLP, hence, it is difficult to solve. Next, we develop a new efficient algorithm to solve this problem.

IV. PROPOSED FEDERATED DDQN ALGORITHM

When addressing the joint optimization problem in multi-user scenarios, various challenges come to light.

- The significant mobility observed in IoT devices results in frequent and unpredictable shifts within the communication channel. This dynamic variability poses a considerable challenge for IoT devices, impeding their ability to effectively make optimal real-time decisions.
- Due to constrained resources, the decision made by each individual IoT device holds a consequential influence on the selection and actions of other devices within the network, meaning that optimal decision-making by one device not only affects its own resource

Algorithm 1 Proposed FDRL for Joint Offloading Decision and Computing and Communication Resource Allocation (Training Phase)

Input: $\mathcal{N}$, $\mathcal{M}$, $t = 0$
Output: $\mathbf{P}, \mathbf{f}^{loc}, \mathbf{f}^{edge}, \mathbf{x}$

Initialize the global model ϕ^{global}, the online and target networks for each agent m.
while (maximum number of iterations is not reached) **do**
  $\phi_m^{\text{online}} = \phi_m^{\text{target}} = \phi_m^{\text{global}}$,
  Using (13), choose the set of participating devices, $\mathcal{K}$.
  for each device m in $\mathcal{K}$ **do**
    for each time step t if $|\mathcal{I}_m| > 0$ **do**
      Compute offloading decision, $\mathbf{x}$, using ϕ_m^{online},
      Interact with environment and solve (9) or (10) applying KKT conditions,
      Save the experience $\mathbf{P}$, $\mathbf{f}^{loc}$ and $\mathbf{f}^{edge}$ in replay memory $\mathcal{O}_{m,n,t}$,
      Train the local model on $\mathcal{O}_{m,n,t}$,
      Transmit ϕ_m^{online} to the PS,
    end for
  end for
  update ϕ_m^{global} using (14) and distribute it to all selected devices in the next round.
end while

allocation and performance but also ripples through the network, influencing the decisions and performance of other devices.

- As the number of IoT devices grows in the network, the offloading decision complexity increases exponentially. This escalating complexity underscores the need for scalable and efficient approaches to address the evolving demands of IoT networks.

To address the issues mentioned above, we propose a multi-agent DDQN algorithm to solve the secure offloading and computing and communication resource allocation problem. The details of the algorithm are elaborated in the following section.

A. Sketch of double deep Q-network

Given the aforementioned challenges, employing traditional optimization methods to address the dynamic optimization problem described in equation (8) is not feasible. Model-free DRL is a useful tool for handling the DMP and learning the optimal solutions in dynamic environments. Hence, the DMP is formulated as a Markov Decision Process (MDP). Particularly, we model our problem as a multi-agent DDQN problem. For each device (DRL agent), we have following components:

- *State space:* the state space for each agent m, denoted by s_m, is defined as $s_m = \{\mathcal{I}_m, h_{m,n}, I_m, c_m, E_{\max}, F_{\max}\}$, where $\mathcal{I}_m$ is the length of the task queue of device m.
- *Action space:* The action space of agents, denoted by $\mathcal{A}$, includes the offloading decisions depending on the current state.
- *Cost function:* The objective function defined in (8) depends on the value of P_m and $f_{m,n}^{edge}$ when offloading to edge server and f_m^{loc} if local computation is selected. Hence, in order to accurately capture the benefits of a specific offloading decision within the cost function, it is imperative to carefully optimize these variables. Therefore, when device m performs its task locally, i.e., $x_{m,n} = 0$, the cost would be calculated by solving the following optimization problem:

$$\min_{f_m^{loc}} \tau_m^{loc} + \lambda E_m^{loc} \quad \text{subject to: C1, C2, C3.} \tag{9}$$

If device m offloads its task to the BS, i.e., $x_{m,n} = 1$, the transmit power and computing frequency at the edge server would be optimized by solving the following optimization problem:

$$\min_{P_m, f_{m,n}^{edge}} (\tau_{m,n}^{comp} + \tau_{m,n}^{air}) + \lambda E_{m,n}^{air} \quad \text{subject to: C1, C2, C4.} \tag{10}$$

It's important to note that both equations (9) and (10) represent convex optimization problems with respect to the variables f_m^{loc} and P_m, $f_{m,n}^{edge}$, respectively. These types of optimization problems can be effectively solved using standard software tools, e.g., Karush-Kuhn-Tucker (KKT) conditions. After finding the optimal computing and communication resource allocation vectors, $\mathbf{f}^{loc}$, $\mathbf{P}$ and $\mathbf{f}^{edge}$, we feed them into the DDQN framework as the immediate cost function. After the DDQN agent is trained through this process for one training round, we apply a FL framework where each IoT device will train its DDQN models, share and update their models with PS.

TABLE I
SIMULATION PARAMETERS

Notation	Parameter	Value
N	Number of BSs	5
M	Number of IoT devices	30
P_m	Transmit power of a IoT device	10 mW
G_n,G_m	Antenna gains	1
f_{max}	Computing frequency	30 GHz
N_0	Noise power spectral density	-90 dBm/Hz
ω	Total system bandwidth	50 MHz
τ_{th}	Service latency requirement	10-100 ms [1]

B. DDQN training phase

Consider the immediate cost of each IoT device m obtained from the solution of the (9) and (10) as $cost_m(s, a)$. Using Bellman equation, the action-state value is:

$$Q_m(s,a) = cost_m(s,a) + \gamma \sum_{s' \in \mathcal{S}} W_{ss'}(a) \max_{a' \in \mathcal{A}} Q_m^*(s', a'), \tag{11}$$

where $\mathcal{S}, W_{ss'}(a)$, and $0 < \gamma < 1$ are the set of states, the transition probability function, and the discount factor, respectively. Our goal by using DDQN is to find a solution to minimize the state-action function $Q_m^*(s', a')$. Each agent m has two neural networks working alongside each other, one called *online network* with parameters ϕ_m^{online} and the other called *target network* with parameters ϕ_m^{target}. At each training iteration the target value for training the online network in device m is calculated as:

$$Z_m = cost_m(s,a) + \gamma Q_m(s', \max_{a' \in \mathcal{A}} Q_m^*(s', a'; \phi_m^{\text{online}}), \phi_m^{\text{target}}) \tag{12}$$

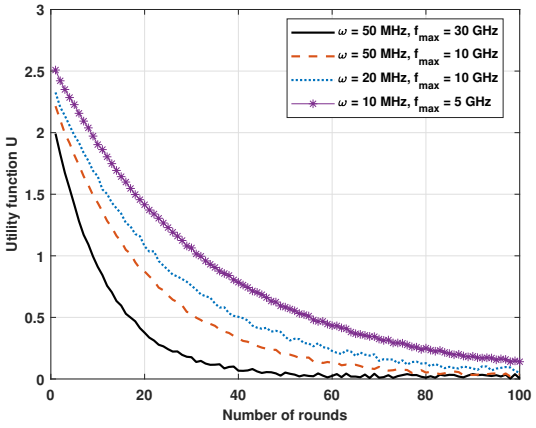

Fig. 2. The convergence property of the FDRL algorithm.

To overcome related issues regarding to train a DRL agent in a centralized manner such as scalability and privacy and to improve the overall performance gains, we propose FDRL that empowers each agent to train its own local model, using its own local data. Then these local models are sent to a PS to be combined together. This process continues until a certain criterion is met.

C. Federated DRL Scheme

Training FDRL agents is completed in three steps. First, small subset of M IoT devices, denoted by $\mathcal{K} = 1, ..., K$ is selected to contribute in FL based on the following criterion:

$$\max_{m \in \mathcal{M}} \text{Variance} \left(\frac{d_m P_{\max}}{F_{\max}} \right), \tag{13}$$

where d_m represents the distance of device from BS. Then, each selected IoT devices use DDQN to train their local models and send the weights of online network, ϕ_m^{online}, to the PS. Finally, the PS aggregates the models using FedAvg [24] and forms a single global model that would be then transmitted to all selected devices. The model aggregation is given as:

$$\phi^{\text{global}} = \frac{\sum_{m \in \mathcal{K}} \phi_m^{\text{online}}}{|\mathcal{K}|} \tag{14}$$

The proposed algorithm is summarized in Algorithm 1.

V. Simulation Results

In this section, we evaluate the performance of the proposed scheme for reliability optimization in the MEC network shown in Fig. 1. The packet size of each IoT devices is selected randomly from a uniform distribution with a range [1, 10] kbits. Simulation parameters are summarized in Table I. We compare the performance of the proposed method with two baseline approaches. The first baseline approach, hereinafter referred to as "Baseline 1", uses simple distributed DDQN without any aggregation (non-federated) to solve the proposed problem. The second baseline, hereinafter referred to as "Baseline 2", is the state-of-art centralized deep Q-network (DQN) scheme where there is a centralized entity that makes decisions based on the collective information from all agents. The performance was evaluated by averaging the results over sufficiently large Monte Carlo runs.

Figure. 2 shows the convergence of the proposed FDRL algorithm versus the number of epochs for different system parameters. From Fig. 2, we observe that the proposed algorithm has a fast convergence rate, and can converge within reasonably small number of epochs for all the simulated cases.

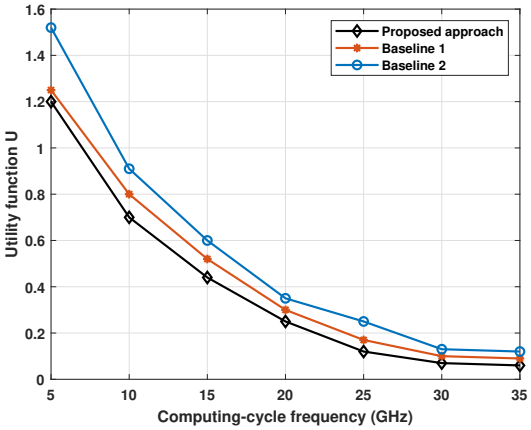

Fig. 3. Utility function versus the computing frequency.

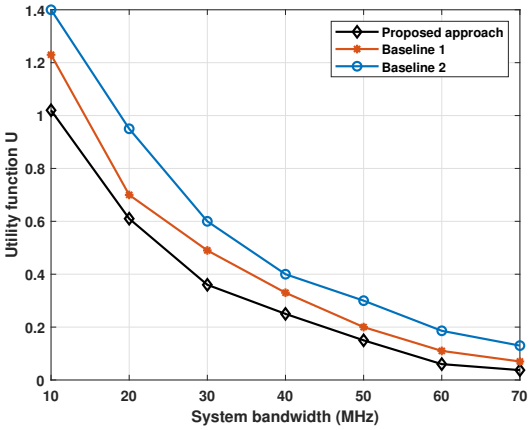

Fig. 4. Utility function versus the bandwidths.

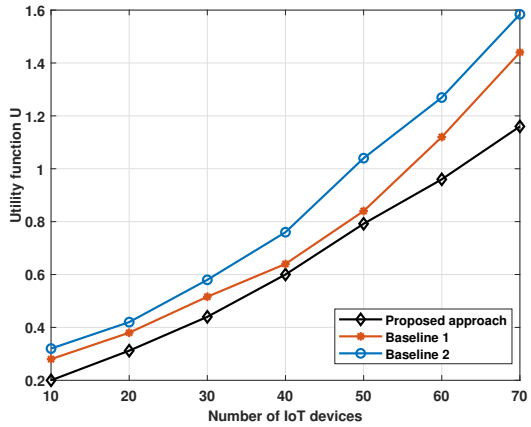

Fig. 5. Utility function versus the number of IoT devices.

Figure. 3 compares the utility function for the proposed approach with the baseline methods, versus the computing-cycle frequency of each MEC. Simulation results show that the utility function declines as the computing capacity increases for all three methods. The results in Fig. 3 also indicate that the proposed scheme surpasses the other two methods. For example, for f^{edge} = 25 GHz , the performance gain is up to 24% and 38%, respectively, compared to baseline schemes 1 and 2 when N = 5 BSs and M = 30 IoT devices.

Figure. 4 illustrates how the utility function is affected by the allocation of different bandwidth. As the bandwidth increases, the utility function decreases due to the fact that each IoT device requires to dedicate small amount of power to achieve higher transmission rates. Therefore, both delivery latency and energy consumption are reduced. The results in Fig. 4 indicate the superior performance of the presented scheme compared to the baseline methods. For instance, in a MEC network with assigned bandwidth ω = 40 MHz, the performance gains yielded by the proposed algorithm are up to 15% and 27%, respectively, compared to baseline methods 1 and 2.

In Fig. 5, the utility function versus the network size is shown for the proposed approach and the two baseline methods. It is clear that the utility function increases as more IoT devices exist in the network. The results in Fig. 5 highlights that the proposed scheme can yield up to 13% and 30% performance gain, when M = 40, compared to the baseline 1 and 2, respectively. Furthermore, Fig. 5 also shows the scalability of the proposed method. For example, with the utility function of 0.9, the proposed scheme can support up to 60 IoT devices, which is 18% and 33% higher compared to baselines 1 and 2, respectively.

VI. Conclusions

In this paper, we addressed a joint latency and energy minimization problem for the NOMA-assisted MEC IoT network to guarantee the reliability of the network. Considering the strict QoS requirement for the IoT devices, the problem was formulated by jointly optimizing the offloading decisions, computing recourse allocation and transmit power control. To solve the proposed MINLP, we have developed a new algorithm that adopted FL, DDQN, and optimization theory called FDRL. More specifically, DDQN is used to determine the offloading decisions of the IoT devices. Given the offloading decisions, the computing capacity or transmit power of the devices is optimized to minimize the utility function. Then, we feed the results into the DDQN framework as the immediate cost function to optimize the offloading decisions. To improve the learning speed of IoT devices, we use FL at the end of each round. Simulation results have confirmed the effectiveness of the proposed scheme to those comparative algorithms.

References

[1] O. Semiari, W. Saad, M. Bennis, and M. Debbah, "Integrated millimeter wave and sub-6 GHz wireless networks: A roadmap for joint mobile broadband and ultra-reliable low-latency communications," *IEEE Wireless Communications*, vol. 26, no. 2, pp. 109–115, 2019.

[2] A. Ahmadi and O. Semiari, "Reinforcement learning for optimized beam training in multi-hop terahertz communications," in *ICC 2021-IEEE International Conference on Communications*. IEEE, 2021, pp. 1–6.

[3] 3GPP, "Study on scenarios and requirements for next generation access technologies," *Technical Report (TR)* 38.913, *Version* 15.0.0, 2018.

[4] Q. Pham et al., "A survey of multi-access edge computing in 5G and beyond: Fundamentals, technology integration, and state-of-the-art," *IEEE Access*, vol. 8, pp. 116974–117017, 2020.

[5] Y. Mao, C. You, J. Zhang, K. Huang, and K. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[6] A. Khalili, Sh. Zarandi, and M. Rasti, "Joint resource allocation and offloading decision in mobile edge computing," *IEEE Communications Letters*, vol. 23, no. 4, pp. 684–687, 2019.

[7] A. Ahmadi, O. Semiari, M. Bennis, and M. Debbah, "Variational autoencoders for reliability optimization in multi-access edge computing networks," in *2022 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2022, pp. 752–757.

[8] A. Kovalenko, R. Hussain, O. Semiari, and M. Salehi, "Robust resource allocation using edge computing for vehicle to infrastructure (V2I) networks," in *2019 IEEE 3rd International Conference on Fog and Edge Computing (ICFEC)*. IEEE, 2019, pp. 1–6.

[9] U. Saleem, Y. Liu, S. Jangsher, X. Tao, and Y. Li, "Latency minimization for d2d-enabled partial computation offloading in mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4472–4486, 2020.

[10] E. Mustafa, J. Shuja, K. Bilal, S. Mustafa, T. Maqsood, F. Rehman, and A. Khan, "Reinforcement learning for intelligent online computation offloading in wireless powered edge networks," *Cluster Computing*, vol. 26, no. 2, pp. 1053–1062, 2023.

[11] X. Li, L. Lu, W. Ni, A. Jamalipour, D. Zhang, and H. Du, "Federated multi-agent deep reinforcement learning for resource allocation of vehicle-to-vehicle communications," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 8, pp. 8810–8824, 2022.

[12] J. Xu, B. Ai, L. Chen, Y. Cui, and N. Wang, "Deep reinforcement learning for computation and communication resource allocation in multiaccess mec assisted railway iot networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 23797–23808, 2022.

[13] A. Taleb Zadeh Kasgari, W. Saad, M. Mozaffari, and H. V. Poor, "Experienced deep reinforcement learning with generative adversarial networks (gans) for model-free ultra reliable low latency communication," *IEEE Transactions on Communications*, vol. 69, no. 2, pp. 884–899, 2020.

[14] S. Wang, M. Chen, X. Liu, Ch. Yin, Sh. Cui, and H. V. Poor, "A machine learning approach for task and resource allocation in mobile-edge computing-based networks," *IEEE Internet of Things Journal*, vol. 8, no. 3, pp. 1358–1372, 2020.

[15] J. Park, S. Samarakoon, M. Bennis, and M. Debbah, "Wireless network intelligence at the edge," *Proceedings of the IEEE*, vol. 107, no. 11, pp. 2204–2239, 2019.

[16] H Brendan McMahan et al., "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1, 2021.

[17] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečnỳ, S. Mazzocchi, H. McMahan, et al., "Towards federated learning at scale: System design," *arXiv preprint arXiv:1902.01046*, 2019.

[18] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah, "Distributed federated learning for ultra-reliable low-latency vehicular communications," *IEEE Transactions on Communications*, vol. 68, no. 2, pp. 1146–1159, 2019.

[19] H. Chen, Sh. Huang, D. Zhang, M. Xiao, M. Skoglund, and H. V. Poor, "Federated learning over wireless iot networks with optimized communication and resources," *IEEE Internet of Things Journal*, vol. 9, no. 17, pp. 16592–16605, 2022.

[20] J. Pei, Sh. Li, Zh. Yu, L. Ho, W. Liu, and L. Wang, "Federated learning encounters 6g wireless communication in the scenario of internet of things," *IEEE Communications Standards Magazine*, vol. 7, no. 1, pp. 94–100, 2023.

[21] J. Mills, J. Hu, and G. Min, "Communication-efficient federated learning for wireless edge intelligence in iot," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5986–5994, 2019.

[22] Y. Han, D. Li, H. Qi, J. Ren, and X. Wang, "Federated learning-based computation offloading optimization in edge computing-supported internet of things," in *Proceedings of the ACM Turing Celebration Conference-China*, 2019, pp. 1–5.

[23] M. M. K. Tareq, O. Semiari, M. Salehi, and W. Saad, "Ultra reliable, low latency vehicle-to-infrastructure wireless communications with edge computing," in *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2018, pp. 1–7.

[24] B. McMahan, E. Moore, D. Ramage, S. Hampson, and A. Blaise, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.