IEEE INFOCOM 2025 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) | 979-8-3315-4370-9/25/$31.00 ©2025 IEEE | DOI: 10.1109/INFOCOMWKSHPS65812.2025.11152979

A Lyapunov-Guided Task Offloading Approach for Backscatter Communication Assisted Edge Computing

Huaming Wu*, Junhui Du*, Huijun Tang†, Pengfei Jiao† and Ruidong Li‡
*Center for Applied Mathematics, Tianjin University, Tianjin 300072, China
†School of Cyberspace, Hangzhou Dianzi University, Hangzhou 310018, China
§Institute of Science and Engineering, Kanazawa University, Kanazawa 920-1192, Japan
Email: {whming, dujunhui_0325}@tju.edu.cn, {tanghuijune, pjiao}@hdu.edu.cn, liruidong@ieee.org

***Abstract*—Optimizing multi-user Mobile Edge Computing (MEC) networks, particularly those with time-varying wireless channels and Backscatter communication (BackCom) models, presents a significant challenge. To address this, we propose an online offloading algorithm that maximizes system data throughput while ensuring stable long-term average energy consumption. This optimization problem is formulated as a multi-stage stochastic Mixed Integer Non-Linear Programming (MINLP) problem, where both binary offloading decisions and resource allocation across multiple time slots are jointly optimized. To handle the complexity introduced by coupled decision-making at different time slots, we introduce LyDRL, a novel approach that combines Lyapunov optimization with Deep Reinforcement Learning (DRL). LyDRL is further enhanced with a threshold quantization method, which significantly reduces computational time and is well-suited for real-time implementation, particularly in environments where channel fading is rapid and unpredictable. Simulation results show that LyDRL reduces runtime by nearly 50% compared to state-of-the-art approaches, confirming its effectiveness and efficiency in dynamic MEC networks.**

***Index Terms*—Mobile Edge Computing, Backscatter Communication, Task Offloading, Lyapunov Optimization, Energy Harvesting**

I. Introduction

MOBILE Edge Computing (MEC) technology is recognized as a crucial solution for enhancing the computational efficiency of Mobile Devices (MDs), especially for Internet of Things (IoT) devices, which often face limitations in battery capacities and processing power [1], [2]. By offloading computational and automation tasks to edge nodes closer to the data source, resource-constrained devices can significantly reduce energy consumption [3], [4]. However, optimizing a multi-user MEC network presents challenges, particularly in the context of time-varying wireless channels, which can complicate resource allocation and network performance.

Backscatter communication (BackCom) is a promising next-generation communication technique that supports wireless connectivity for sensors and actuators, enables seamless data exchange between physical and digital components, and enhances energy efficiency in resource-constrained environments. BackCom separates the transmitter into two parts: one responsible for emitting the Radio-Frequency (RF) carrier, which includes power-intensive elements such as synthesizers and amplifiers, and the other containing the information, equipped with energy-efficient components like harvesters and modulators.

To significantly extend the battery life of MDs while optimizing spectrum utilization, this paper integrates BackCom technology into MEC. Unlike traditional Wireless Power Transfer (WPT) approaches, the BackCom-based scenario enables MDs to modulate received energy signals, allowing them to transmit a portion of their task data during offloading to the MEC server. This not only enhances the durability and operational lifespan of MDs. but also improves overall system efficiency. Furthermore, to further boost the performance of the BackCom model, we incorporate a non-orthogonal multiple access (NOMA) communication model, we introduce a Non-Orthogonal Multiple Access (NOMA) communication scheme, which allows multiple MDs to simultaneously communicate with MEC servers using the same spectral resources. We explore a multi-user MEC network assisted by a BackCom model and design an online computation offloading algorithm that maximizes system data throughput under long-term average power stability constraints. To address this challenge, we propose an online Lyapunov-guided task offloading algorithm called LyDRL, specifically tailored for NOMA-based and Wireless-Powered MEC with BackCom. This algorithm leverages the strengths of Lyapunov optimization and Deep Reinforcement Learning (DRL) [5]. The primary contributions of this paper are summarized as follows:

- To address the energy limitation of MDs, both WPT-assisted MEC and backscatter-assisted MEC can improve the endurance of MDs. This paper integrates these two models within a binary offloading decision framework in edge computing. Specifically, when the MD opts for local processing, WPT is utilized to extend the MD's battery life; when the MD decides to offload tasks to the MEC

This work is supported by the National Natural Science Foundation of China under Grant Number 62071327 and 62401190. Corresponding author: Huijun Tang.

server, the backscatter method is employed.

- We have introduced modifications to the data transmission and energy harvesting components of the backscatter model. To enhance spectrum utilization, we employ the NOMA model in the data transmission phase, allowing for more efficient communication among multiple MDs. In the energy harvesting phase, we adopt a nonlinear energy harvesting model to more accurately reflect real-world energy harvesting dynamics, providing a more realistic and effective approach to energy collection.
- We design an innovative order-preserving threshold quantization technique. This method effectively balances the exploration-exploitation trade-off in the design of the DRL algorithm, ensuring swift training convergence. This approach effectively reduces time complexity while maximizing system data throughput.

The remainder of this paper is organized as follows: In Section II, we provide an overview of related work in this field. The system model can be found in Section III. Section IV formulates the MINLP problem about maximizing system data throughput and provides an efficient algorithm for making offloading decisions and resource allocation. Simulation results are provided in Section V. Section VI concludes this paper.

II. Related Work

To optimize the computational performance of multi-user MEC networks, extensive research has focused on computation offloading strategies. A significant portion of this research addresses the optimization of computation offloading, often involving Mixed-Integer Nonlinear Programming (MINLP) problems. These problems typically involve jointly determining binary offloading decisions and resource allocation choices. The complexity of such optimization problems primarily arises from the challenge of efficiently optimizing the binary offloading decision variables, making the problem inherently difficult to solve.

Most communication models in MEC systems are based on Time Division Multiple Access (TDMA) [6] or Frequency Division Multiple Access (FDMA) [7]. However, these models often suffer from low communication spectrum utilization. Recently, backscatter-aided MEC networks have gained increasing popularity [8]. For instance, Shi *et al.* [9] focused on optimizing the total computation rate of MDs in a BackCom-MEC system. In their scenario, the computing task is divisible, partial offloading is used, and the MD's computing resources are optimized. Xie *et al.* [10] explored the use of RF communication and BackCom for offloading computational tasks. Additionally, many studies have leveraged Deep Neural Network (DNN) technology to design online offloading algorithms in MEC networks [11]. By exploiting the data fitting capabilities of DNNs, these methods can capture data characteristics from large volumes of historical data, enabling the prediction and classification of new data for future decision-making.

The majority of previous research typically incorporates only one or two technical methodologies. To enhance system performance, this paper adopts an MEC system assisted by the BackCom model. By leveraging the benefits of Lyapunov optimization and DRL, it develops a more robust algorithmic model suited for this particular scenario.

III. System Model

As illustrated in Fig. 1, we consider an edge computing system based on the BackCom model, consisting of a Power Beacon (PB), an Access Point (AP) equipped with a MEC server, and K MDs. Each MD is equipped with an Energy Harvesting (EH) circuit or a WPT circuit, a BackCom circuit, and a local processor. We assume that each circuit and processor operates independently, allowing each MD to backscatter task data without any interference. In addition to locally processing task data, each MD can simultaneously offload and process task data while harvesting energy, thanks to the independence of the circuits.

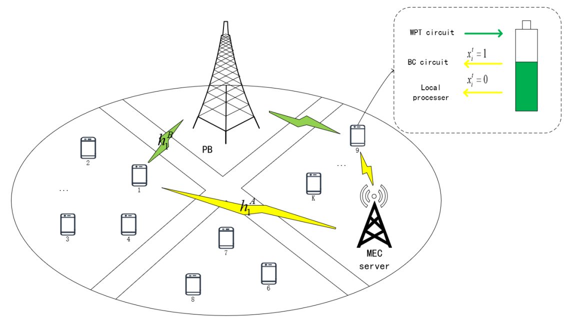


Fig. 1: An overall BackCom-assisted MEC network

The system's operation time is divided into discrete time slots, each with a duration of τ. During the t-th time slot, let $h_i^B[t]$ and $h_i^A[t]$ denote the channel gains from the PB to the i-th MD and from the i-th MD to the AP during the t-th time slot, respectively, where $i \in \{1, 2, \ldots, K\}$. It's worth noting that, in this paper, all the channel models are assumed to be quasi-stationary [9], meaning that the channel conditions remain constant within each time slot but may vary between consecutive time slots.

Additionally, we assume that the Access Point (AP) has perfect knowledge of the Channel State Information (CSI) for all involved channels. While obtaining accurate CSI is typically challenging in real-world deployments, the proposed approach can be viewed as an idealized simulation that closely approximates real-world conditions. This setup provides an upper bound on the model's performance, offering valuable insights into its potential in practical scenarios. To simplify the subsequent calculation of the data throughput for different MDs, we assume that the communication channel gains are sorted in descending order based on the MD's ordinal number.

A. Task Offloading and Energy Harvesting Model

This paper primarily focuses on indivisible computational tasks. We use the binary variable x_i^t to represent the offloading decision of the i-th MD in the t-th slot.

When $x_i^t = 0$, it corresponds to a scenario similar to the standard WPT process. However, in this context, both energy harvesting and local processing can occur simultaneously, allowing for enhanced system performance. We define the CPU frequency of the i-th MD's local processor in the t-th time slot as f_i^t. Since each circuit and device operate independently, the amount of data that the MD can process within one time slot is: $D_{i,L}^t = \frac{f_i^t \tau}{C_{cpu}^i}$, where C_{cpu}^i represents the CPU cycles required by the i-th MD's local processor to process 1 bit of data. The energy consumed by the MD within this time slot can be expressed as: $E_{i,L}^t = k(f_i^t)^3 \tau$, where k is the Energy Consumption Coefficient (ECC) of the processor chip on the MD.

Previous studies typically use a linear energy harvesting model, where energy received by each MD is proportional to the PB's emission power. However, this model overlooks the nonlinear characteristics of the MD's energy harvesting circuit, leading to performance degradation. To address this, we propose a nonlinear energy harvesting model that better reflects the circuit's actual behavior. The amount of energy that i-th MD can harvest in the current time slot is $E_{i,L}^H = \left(\frac{c_i P_t h_i^B[t] + d_i}{P_t h_i^B[t] + v_i} - \frac{d_i}{v_i}\right)\tau$, where c_i, d_i and v_i are parameters characterizing the nonlinear energy harvesting model of the i-th MD, and P_t is the energy emission power of PB [12].

When $x_i^t = 1$, the BackCom technique is employed for task offloading. To enhance communication performance, the NOMA communication mode is used to efficiently utilize spectrum resources in conjunction with BaFckCom. In this scenario, MDs opting to offload tasks to the MEC server experience simultaneous backscatter and wireless charging. The i-th MD divides the energy signal emitted by the received PB into two parts based on the backscatter coefficient a $(0 \le a \le 1)$. A portion of the energy signal, aP_t, is used to backscatter part of the mission data to the MEC server via the upstream NOMA, while the remaining portion, $(1-a)P_t$, is used to charge the MD and replenish its energy reserves.

In the t-th time slot, given the CSI between the MD and the AP, as well as the PB, the total data throughput of all MDs selected with $x_i^t = 1$ can be calculated as follows:

$$\sum_{x_i^t=1} D_i^t = \tau B \log_2\Big(1 + \sum_{x_i^t=1} \frac{aP_t H_i^t}{\sigma^2}\Big), \tag{1}$$

where $H_i^t = h_i^B[t] \times h_i^A[t]$.

The energy that the i-th MD can harvest in the t-th time slot is $E_{i,o}^t = \left(\frac{c_i(1-a)P_t h_i^B[t] + d_i}{(1-a)P_t h_i^B[t] + v_i} - \frac{d_i}{v_i}\right)\tau$. In this phase, due to the inherent circuit consumption of the BackCom circuit, we express the inherent circuit power consumption of the BackCom circuit as $E_{i,o}^t = P_s \tau$. Due to the small amount of data returned by the task under normal circumstances, and the powerful computing power of the MEC server, similar to [13], we ignore both the processing time of the MEC server and the return time for the MD's computational results.

IV. Problem Formulation And Solution

This paper aims to design an online algorithm that maximizes the long-term average weighted computational throughput of all MDs, while maintaining stability in the long-term average energy consumption. To achieve this, we optimize various aspects within each time slot, including offload decisions denoted as $X^t = [x_1^t, x_2^t, \cdots, x_K^t]$, the backscatter factor a, the CPU frequency $f^t = [f_1^t, f_2^t, \cdots, f_K^t]$ for the local processor. Specifically, when $x_i^t = 0$, we set $a = 0$, and when $x_i^t = 1$, we assign $f_i^t = 0$. This problem is conceptualized as a multi-stage stochastic MINLP problem:

$$\boldsymbol{\mathcal{P}_1}: \max_{\{X^t\}_{t=1}^T, a, \{f^t\}_{t=1}^T} \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{K} w_i r_i^t \tag{2}$$

$$s.t. \quad \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left(Q_i[t]\right) < \infty; \tag{2a}$$

$$0 \le a \le 1; \tag{2b}$$

$$0 \le f_i^t \le f_i^{\max}, \forall i; \tag{2c}$$

$$x_i^t \in \{0, 1\}, \forall i; \tag{2d}$$

where r_i^t denotes the computational workload achieved by the i-th MD during the t-th time slot: $\sum_{i=1}^K r_i^t = \sum_{x_i^t=0} D_{i,L}^t + \sum_{x_i^t=1} D_i^t$. To address the long-term average energy consumption stability constraint, a model is introduced for the MD's energy consumption queue [13], as follows: $Q_i[t+1] = \max\left\{Q_i[t] - E_i^H + E_i^t, 0\right\}$, where $E_i^H = x_i^t E_{i,o}^H + (1-x_i^t) E_{i,L}^H$ and $E_i^t = x_i^t E_{i,o}^t + (1-x_i^t) E_{i,L}^t$. $Q_i[t]$ represents the energy consumption of the i-th MD in the t-th time slot, which can be seen as a queue with "energy arrival", while E_i^H behaves as a queue with a "service rate". When the energy queue achieves stability, constraint (2a) is satisfied. Constraint (2b) restricts the allowable range of values for the backscatter factor; Constraint (2c) ensures that the CPU calculation frequency of the local processor of the i-th MD does not exceed the local maximum CPU calculation frequency; Constraint (2d) represents that the binary offloading decision factor must adhere to certain conditions.

A. Lyapunov Optimization-based Multi-stage MINLP Problem

In this section, we employ Lyapunov optimization theory to decouple constraint (2a) into individual deterministic constraints for each time frame. We define the Lyapunov function as $L(Q[t]) = \frac{1}{2}\sum_{i=1}^{K} Q_i^2[t]$, and the Lyapunov drift $\Delta L(Q[t])$ as: $\mathbb{E}\{L(Q[t+1]) - L(Q[t]) | Q[t]\}$, where $Q[t] = \{Q_i[t]\}_{i=1}^K$. Let's find the upper bound of $\Delta L(Q[t])$, which can be calculated as follows: $\Delta L(Q[t]) \le Q_1 + \sum_{i=1}^{K} Q_i[t] \mathbb{E}\left[\left(E_i^t - E_i^H\right) | Q_i[t]\right]$.

Proposition 1: As the transmit power of the PB continues to increase, the energy harvesting power of the MD also increases until it reaches a convergence point.

Proof: First, let us define the function $F(\alpha)$ as follows:

$$F(\alpha) = \frac{c_i(1-a)P_t h_i^B[t] + d_i}{(1-a)P_t h_i^B[t] + v_i} - \frac{d_i}{v_i} \ (0 \le a \le 1),$$

where we introduce a variable $y = (1-a)P_t h_i^B[t]$ to represent the transmit power of the PB. Importantly, $F(y)$ exhibits monotonic behavior with respect to y and remains unaffected by the sign of the first-order derivative of $F(y)$.

Furthermore, considering the range of possible channel values, we can establish that $\max F(y) = \frac{c_i P_t + d_i}{P_t + v_i} - \frac{d_i}{v_i}$. Similarly, for the expression $\frac{c_i P_t h_i^B[t] + d_i}{P_t h_i^B[t] + v_i} - \frac{d_i}{v_i}$. We can deduce that its maximum value is also $\max F(y)$. ■

According to Proposition 1, with respect to Q_1 we obtain:

$$Q_1 \triangleq \frac{1}{2} \sum_{i=1}^{K} \left[\left(\left(\frac{c_i P_t + d_i}{P_t + v_i} - \frac{d_i}{v_i} \right) \tau \right)^2 + \max \left\{ P_s \tau, k \left(f_i^t \right)^3 \tau \right\} \right]. \tag{3}$$

In the t-th time slot, we apply opportunity expectation minimization techniques to decouple long-term energy consumption into individual time slots for processing. For the original problem, the objective function seeks to maximize the amount of long-term data computation. This can be achieved by maximizing the amount of MD data computation in each time slot as much as possible. To summarize, we decompose the problem $\boldsymbol{\mathcal{P}_1}$ into deterministic optimization problem $\boldsymbol{\mathcal{P}_2}$ for each time frame:

$$\boldsymbol{\mathcal{P}_2}: \min_{X^t, a, f^t} \sum_{i=1}^{K} Q_i[t] \left(E_i^t - E_i^H \right) - V \sum_{i=1}^{K} w_i r_i^t \tag{4}$$

$$s.t. \ (2b); \ (2c); \ (2d); \tag{5}$$

where V is used to measure the "degree" of punishment. It is evident that because of the discontinuity of offloading decision factor x_i^t, certain conventional convex optimization methods may not effectively address the optimization problem $\boldsymbol{\mathcal{P}_2}$. Generally speaking, obtaining an approximate solution often requires resorting to convex relaxation or near-convex optimization methods. However, these methods tend to have high computational complexity, and deriving an approximate solution can be challenging.

B. Lyapunov Optimization-Aided DRL

In order to determine the optimal value in the t-th time slot, each MD primarily observes a partial system state denoted as $S^t = \left\{ d_i^A[t], d_i^B[t], h_i^A[t], h_i^B[t], Q_i[t] \right\}_{i=1}^{K}$. This state includes information on the distances between the i-th MD and the AP and PB in the t-th time slot, the channel states between the MD and the AP and PB, and the energy consumption queue data for each MD. Additionally, the corresponding model needs to provide binary offloading decisions and resource allocation actions $A^t = \{x_i^t, y_i^t\}_{i=1}^{K}$, where $y_i^t = \{f_i^t, a\}_{i=1}^{K}$ represents the result of resource allocation. If we denote $Y\left(\{x_i^t\}_{i=1}^{K}, S^t \right)$ as the optimal value of problem $\boldsymbol{\mathcal{P}_2}$ given $\{x_i^t\}_{i=1}^{K}$ and observed state S^t, then the selection of $\{x_i^t\}_{i=1}^{K}$ can be determined by:

$$X_{best}^t = \underset{X^t \in \{0,1\}^K}{\arg\max} \ Y \left(\left\{ x_i^t \right\}_{i=1}^{K}, S^t \right), \tag{6}$$

where $X^t = [x_1^t, x_2^t, \cdots, x_K^t]$ represents the offloading decision of all MDs in the t-th time slot.

We leverage the powerful capabilities of DRL to develop a model with reduced complexity. which is designed to identify the best course of action by considering different states, making it well-suited for scenarios with frequent changes in channel states and other dynamic conditions. The algorithm proposed in this paper consists of the following key modules:

1) Actor Module: The performer module comprises primarily two components: a neural network and an action quantizer. At the outset of the current time slot, the neural network's parameters are initialized as θ^t, following a standard Gaussian distribution. Subsequently, the neural network receives a set of state inputs S^t, and then outputs a continuous offloading decision after passing through the fully connected layer:

$$\Pi_{\theta^t}: S^t \to X^t = \left\{ X_i^t \in [0,1], i = 1, 2, \cdots, K \right\}. \tag{7}$$

Following the passage through the action quantizer, the continuous value is quantized into discrete binary offloading decisions. Building on the advantages of order-preserving quantization [13], we propose a novel multi-threshold sequence-preserving quantizer. This quantizer helps alleviate the high complexity associated with greedy algorithms while generating a sufficient number of offloading decisions, thus improving the overall algorithm's performance.

In scenarios involving K MDs, we can establish multiple thresholds in the following manner: $\{1/K, 2/K, \cdots, i/K\}$, $i = \{1, 2, \cdots, K\}$. Each threshold, denoted as i/K, corresponds to a set of candidate offloading decision factors, represented by:

$$x_i^t = \begin{cases} 1 & \text{if } X^t \geq \frac{i}{K} \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

This threshold-based approach produces i candidate offloading decision factors and inherently applies the concept of sequential quantization due to the way thresholds are set. Moreover, implementing the multi-threshold sequence-preserving quantizer in code is more straightforward than using noise-sequencing quantization, making it a more practical solution for real-time system deployment.

2) Critic Module: The critic module primarily focuses on a set of candidate offloading decisions and employs the objective function to filter out the optimal offloading decision.

3) Storage Module: The memory module primarily functions as a storage repository for preserving the optimal offloading decisions that have been selected, along with the corresponding state inputs, as samples within the replay memory.

4) Update Module: The primary role of the update module is to randomly select a set of samples from the replay memory and utilize them to update the network parameters.

C. Sub-Optimization Problems

In the context of two distinct offloading methods, we define M_0 and M_1 as the sets of MDs that choose local computing and those that offload tasks to the MEC server, respectively. This differentiation naturally leads to the decomposition of the optimization problem $\boldsymbol{\mathcal{P}_2}$ into two separate parts, each requiring an individual solution process.

1) Local Processing ($j \in M_0$):

$$\boldsymbol{\mathcal{P}_4}: \min_{f_j^t} Q_j[t]\left[k\left(f_j^t\right)^3\tau - \left(\frac{c_j P_t h_j^B[t] + d_j}{P_t h_j^B[t] + v_j} - \frac{d_j}{v_j}\right)\tau\right] - \frac{V w_j f_j^t}{C_{cpu}^j} \tag{9}$$

$$s.t. \quad 0 \leq f_j^t \leq f_j^{\max}, \forall j; \tag{9a}$$

By setting the derivative of the objective function to zero, we can get $\left(f_j^t\right)^* = \min\left\{\sqrt{\frac{V w_j}{3Q_j(t)k\tau C_{cpu}^j}}, f_j^{\max}\right\}$.

2) Offload to the MEC server ($i \in M_1$): In this context, since data throughput cannot be expressed as a single formula for an individual MD when offloading is performed via BackCom, we choose to directly optimize all MDs using the selected offloading mode.

$$\boldsymbol{\mathcal{P}_5}: \min \sum_{i\in M_1} Q_i[t]\left[P_s\tau - \left(\frac{c_i\left(1-a\right)P_t h_i^B[t] + d_i}{\left(1-a\right)P_t h_i^B[t] + v_i} - \frac{d_i}{v_i}\right)\tau\right] - V w_i B \log_2\left(1 + \sum\nolimits_{i\in M_1} \frac{a P_t H_i^t}{\sigma^2}\right) \tag{10}$$

$$s.t. \quad 0 \leq a \leq 1; \tag{10a}$$

It is evident that the sub-optimization problem is a convex constrained optimization problem. Considering the unique characteristics of the model presented in this paper, which focuses on optimizing a single reflection factor a, we propose a binary search method to reduce the algorithm's complexity. By employing binary search, we can efficiently find the derivative of the sub-optimization objective function, thereby simplifying the computation process. This approach not only reduces the overall complexity but also allows us to obtain a good approximate solution. The detailed step-by-step algorithm is provided in Algorithm 1.

Algorithm 1 LyDRL for $\boldsymbol{\mathcal{P}_2}$.

```
Input: Parameters V, K, training interval δ, T;
Output: offloading factor x_i and y_i^t = {f_i^t, a}_{i=1}^K;
Initialize the DNN parameters;
Empty initial energy consumption queue Q_i(1) = 0;
for i = 1 to T do
    Each MD observes its own state S^t;
    Generate a set of candidate offloading actions x_i via
    DNN and the Action Quantizer;
    For each candidate offloading decision factor x_i; For
    MD in local offload mode, we can obtain the optimal
    local calculation frequency (f_j^t)^*, and then we find the
    optimal a by binary search.
    if mod (t, δ) = 0 then
        Random sampling of samples from memory;
        Update DNN parameters with extracted samples;
    end if
end for
```

V. Performance Evaluation

A. Parameters Setup

We set the AP at coordinates $[0, 0, 10]$ (m) and the PB at coordinates $[50, 10, 5]$ (m). Each time slot is defined to be 1 second long. To ensure a more realistic simulation, we define the movement range of each MD as a circular area centered around the AP with a radius of 3 (m). This approach prevents instances where, over a certain period, MDs move too far from the AP and PB, which could negatively impact system performance. We set the noise power σ^2 to $10^{-20.4}$, β_0 to -20 dB, the ECC of the processor chip on the MD to 10^{-26}, and the channel bandwidth B to be 2 MHz. Additionally, the local maximum calculated frequency $f_i^{\max}$ for the i-th MD is set at 0.3 GHz. The following baselines are selected for comparison:

- **Order-preserving (OP)**: A typical DRL method utilizing sequence-preserving sequence retention quantization [13];
- **OP with noise (OPN)**: An extension of the OP method where noise is added to the quantization process;
- **All Local**: All tasks are processed locally on the MD;
- **All MEC**: All tasks are offloaded and processed on the MEC server;
- **Threshold (Reverse quantization)**: A method that reverses the order of quantization, where values below a certain threshold are quantized as one.

B. Simulation Results

1) Impact of Different Numbers of MDs: As shown in Fig. 2, we analyze the average data throughput of the system under varying numbers of MD. It can be observed that as the number of MDs increases, the total data throughput of the system also increases, but the average data throughput decreases. This is because, as the number of MDs grows, the system gradually approaches its maximum performance capacity, leading to a decrease in the average data throughput.

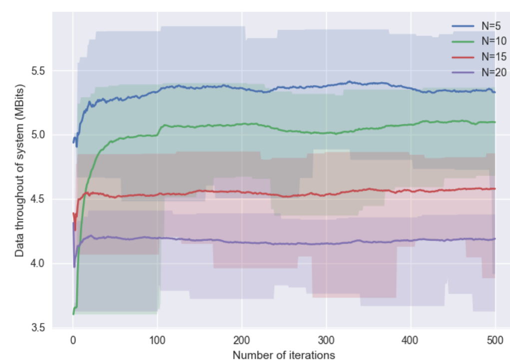
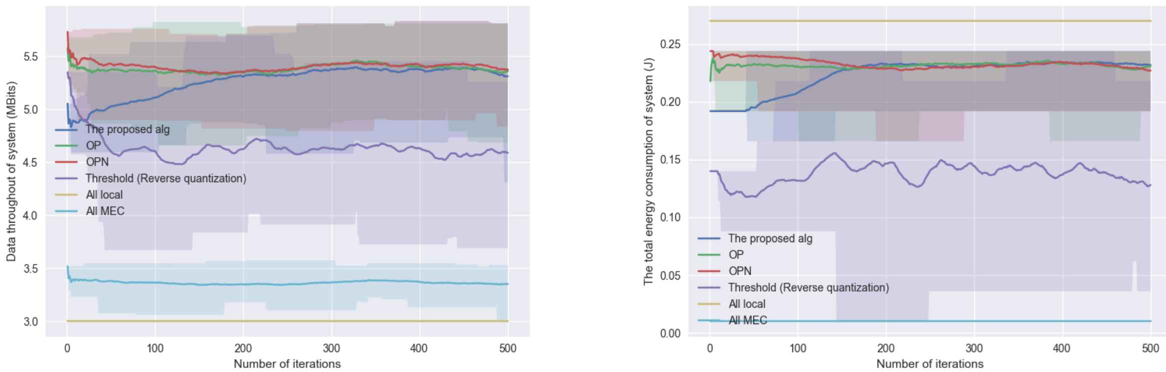


Fig. 2: Average data throughput of the system under different numbers of MD

2) Running Time Comparison: As shown in Table I, the runtime of LyDRL is significantly improved compared to the *OP* and *OPN* schemes. The most substantial improvement is observed when LyDRL is compared to *OPN*. This improvement is particularly important because *OPN* requires additional noise to the output of the DNN model, whereas LyDRL avoids this overhead, resulting in enhanced efficiency.

3) Comparison with Different Algorithms: As illustrated in Fig. 3, when compared to the *All Local* and *All MEC* schemes, the proposed LyDRL exhibits significant advantages



(a) System data throughput

(b) System energy consumption

Fig. 3: System performance under different algorithms.

TABLE I: Comparison of running time of different algorithms

Index	LyDRL	OP	OPN	$\frac{OP}{LyDRL}$	$\frac{OPN}{LyDRL}$
System data throughput (Mbits)	5.4	5.4	5.4	1.0	1.0
CPU computation time (second)	77.71	141.93	305.68	1.82	3.93

in terms of the total average data throughput of the system. Furthermore, in comparison to the *OP* and *OPN* schemes, our method achieves a similar level of system performance. A noteworthy aspect of this approach is its simplicity and ease of implementation. To emphasize this feature, we controlled other variables and only modified the quantization model, recording the time it took for different models to run 500 iterations with the same parameters. This comparison highlights the efficiency and practicality of our proposed method.

We also explored the system's total data throughput trends under varying quantization orders. We established a quantization order wherein values greater than a predefined threshold are quantized to 1. This aligns with the intuitive approach of general order-preserving quantization models. To validate the accuracy of this intuition, we designed the Threshold (Reverse quantization) algorithm. While keeping all other parameters constant, we compared the impact of these two distinct quantization methods on system performance. As depicted in Fig. 3a, it's evident that the system's behavior remains consistent with the chosen quantization method. In summary, our proposed model not only accelerates convergence speed but also delivers commendable system performance, affirming its effectiveness.

VI. Conclusion

In this paper, we propose the LyDRL algorithm, which combines the advantages of Lyapunov optimization and DRL to tackle the challenge of online energy consumption stability computation offloading in multi-user MEC networks. Both theoretical and simulation results demonstrate that this approach achieves optimal computation rate performance while satisfying the long-term energy consumption constraints. Moreover, we employ a threshold quantization method for generating online actions, which converges in fewer iterations, resulting in a lower overall time complexity compared to previous noise-order preservation methods, while maintaining consistent system performance. LyDRL has broad applicability in MEC networks, contributing to improved computational efficiency and the robustness of computing performance.

References

[1] H. Wu, J. Chen, T. N. Nguyen, and H. Tang, "Lyapunov-guided delay-aware energy efficient offloading in iiot-mec systems," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 2117–2128, 2023.

[2] Y. Chen, K. Li, Y. Wu, J. Huang, and L. Zhao, "Energy efficient task offloading and resource allocation in air-ground integrated mec systems: A distributed online approach," *IEEE Transactions on Mobile Computing*, vol. 23, no. 8, pp. 8129–8142, 2024.

[3] H. Wu, L. Tian, H. Tang, R. Li, and P. Jiao, "Graph convolutional reinforcement learning-guided joint trajectory optimization and task offloading for aerial edge computing," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2024.

[4] Y. Liang, H. Tang, H. Wu, Y. Wang, and P. Jiao, "Lyapunov-guided offloading optimization based on soft actor-critic for isac-aided internet of vehicles," *IEEE Transactions on Mobile Computing*, vol. 23, no. 12, pp. 14 708–14 721, 2024.

[5] S. Zhang, S. Bao, K. Chi, K. Yu, and S. Mumtaz, "Drl-based computation rate maximization for wireless powered multi-ap edge computing," *IEEE Transactions on Communications*, vol. 72, no. 2, pp. 1105–1118, 2024.

[6] G. Chen, Q. Wu, W. Chen, D. W. K. Ng, and L. Hanzo, "Irs-aided wireless powered mec systems: Tdma or noma for computation offloading?" *IEEE Transactions on Wireless Communications*, vol. 22, no. 2, pp. 1201–1218, 2022.

[7] H. G. Myung, J. Lim, and D. J. Goodman, "Single carrier fdma for uplink wireless transmission," *IEEE vehicular technology magazine*, vol. 1, no. 3, pp. 30–38, 2006.

[8] J. Du, H. Wu, M. Xu, and R. Buyya, "Computation energy efficiency maximization for noma-based and wireless-powered mobile edge computing with backscatter communication," *IEEE Transactions on Mobile Computing*, vol. 23, no. 6, pp. 6954–6970, 2024.

[9] L. Shi, Y. Ye, X. Chu, and G. Lu, "Computation bits maximization in a backscatter assisted wirelessly powered mec network," *IEEE Communications Letters*, vol. 25, no. 2, pp. 528–532, 2021.

[10] Y. Xie, Z. Xu, Y. Zhong, J. Xu, S. Gong, and Y. Wang, "Backscatter-assisted computation offloading for energy harvesting iot devices via policy-based deep reinforcement learning," in *2019 IEEE/CIC International Conference on Communications Workshops in China (ICCC Workshops)*. IEEE, 2019, pp. 65–70.

[11] Y. Liu, H. Yu, S. Xie, and Y. Zhang, "Deep reinforcement learning for offloading and resource allocation in vehicle edge computing and networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 11, pp. 11 158–11 168, 2019.

[12] Y. Chen, N. Zhao, and M. S. Alouini, "Wireless energy harvesting using signals from multiple fading channels," *IEEE Transactions on Communications*, vol. 65, no. 11, pp. 5027–5039, 2017.

[13] S. Bi, L. Huang, H. Wang, and Y.-J. A. Zhang, "Lyapunov-guided deep reinforcement learning for stable online computation offloading in mobile-edge computing networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 11, pp. 7519–7537, 2021.